UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Desenvolvimento de um Sistema de Reconhecimento de Comandos Verbais para Robôs Baseado na Técnica de Redes Neurais Artificiais

Dissertação submetida à Universidade Federal de Santa Catarina

para obtenção do grau de

Mestre em Engenharia Elétrica

Emerson Pereira Raposo

Florianópolis, 15 de Maio de 1997.

Desenvolvimento de um Sistema de Reconhecimento de Comandos Verbais para Robôs Baseado na Técnica de Redes Neurais Artificiais

Emerson Pereira Raposo

Esta dissertação foi julgada para obtenção do título de

Mestre em Engenharia

especialidade **Engenharia Elétrica,**

área de concentração **Controle, Automação e Informática Industrial,**

e aprovada em sua forma final pelo Curso de Pós-Graduação.

Prof. Marcelo Ricardo Stemmer, Dr. Ing.
Orientador

Prof. Adroaldo Raizer, Dr.
Coordenador da Pós-Graduação Eng. Elétrica

Banca Examinadora:

Prof. Marcelo R. Stemmer, Dr. Ing. (Presidente)

Prof. Fernando Mendes de Azevedo, PhD.

Prof. Guilherme Bittencourt, Dr.

Prof. Jorge Muniz Barreto, Dr.

A Deus, meus Pais, Irmãos, Avós e a minha Namorada.

AGRADECIMENTOS

Eu agradeço ao Prof. Marcelo Ricardo Stemmer pela orientação, oportunidade de co-trabalharmos e pela amizade, que contribuiram para o desenvolvimento desta dissertação. Também desejo agradecer ao Prof. Ubiratan Holanda Bezerra por incentivar-me e encaminhar-me à pesquisa científica.

Aos membros da banca examinadora, pelas críticas e sugestões que contribuíram para o aprimoramente deste trabalho.

Ao LCMI, CAPES, FAPEU, PGEEL e aos seus respectivos membros, por sua estrutura, auxílio e apoio financeiro.

Gostaria de agradecer a Andre Bittencourt Leal, Marcelo Tavares Maciel, Ricardo Ferreira Martins e Waleska Nishida, por sua amizade e companheirismo.

Gostaria de agradecer a Deus, aos meus pais José Esteves Raposo e Terezinha Pereira Raposo; aos meus avós João dos Santos Pereira e Guiomar Pinheiro Pereira, Amadís Augusto Raposo e Elizia Esteves Raposo; e aos meus irmãos Heber Pereira Raposo e Michelli Pereira Raposo por minha educação e por darem muito de suas vidas por mim. A Danielle Nishida, minha Namorada, por todo seu amor.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

O reconhecimento de fala tem várias áreas de aplicação: tradução de textos, ditados, interfaces de computadores, serviços automáticos por telefone e aplicações industriais de propósito geral. A principal razão para o sucesso dos sistemas de reconhecimento tem sido demonstrada pelo aumento na produtividade propiciada por estes, que assistem ou substituem operadores humanos. Esta dissertação tem como objetivo o desenvolvimento de um sistema de reconhecimento de fala para interação com um robô.

As redes neurais artificiais surgem como o principal paradigma para o desenvolvimento destes sistemas, já que estas têm como principais características seu paralelismo, capacidade de treinamento, generalização, não linearidade e robustez. Essas vantagens são confirmadas através dos experimentos realizados neste trabalho, no qual comprova-se a importância das redes neurais artificiais para tais aplicações.

ABSTRACT

There are several application areas for speech recognition: translation of texts, dictation, computer interfaces, automatic telephone services and general purpose industrial applications. The success of the speech recognition systems has been demostrated by the increasing productivity provide by them, attending or substituting the human users or operators. This work is focused on the development of a speech recognition system to interact with a robot.

The artificial neural networks emerge as the main paradigm to the development of those systems, since they have parallelism, trainability, generatization, nonlinearity and robustness characteristics. These advantages are confirmed through the experiments performed in this work, where they attest the importance of the artificial neural networks for applications of that kind.

Capítulo 1

INTRODUÇÃO

No mundo moderno, os seres humanos se utilizam cada vez mais de máquinas com o intuito de facilitar suas vidas. Assim, cada vez mais estas máquinas precisarão ter maior autonomia e o interfaceamento entre elas e o ser humano tem que ser mais fácil. Para que um interfaceamento Homem-máquina mais natural seja possível, têm sido estudados mecanismos que analisam, detectam, reconhecem e descrevem padrões em dados numéricos ou sensoriais. O conjunto destas atividades é geralmente denominado reconhecimento de padrões.

Apesar de todos os avanços na área de reconhecimento de padrões, nenhuma técnica utilizada para captura de informações tem maior eficiência que a utilizada pelo Homem. A visão e a fala são tarefas muito complexas, onde as informações não podem ser avaliadas através de fatos isolados, ou mesmo através de grupos de fatos isolados, mas sim através de fatos inter-relacionados.

A capacidade humana de percepção é bastante aguçada para o reconhecimento de padrões. Podemos reconhecer expressões faladas e imagens de forma bastante robusta, mesmo quando sujeitas a grandes distorções.

Além da percepção, a associação do fato percebido a um padrão já armazenado em sua memória é outra grande capacidade humana. A esta tarefa é parte da cognição, que consiste na seleção heurística de características, baseada em experiências anteriores. É bastante difícil, se não impossível, descrever quais atributos específicos são utilizados para a tomada de decisão.

Mesmo que não iguale a capacidade humana, o reconhecimento de padrões por máquinas e computadores tem sido objeto de inúmeras pesquisas e, atualmente, aplicações de grande relevância têm sido desenvolvidas. Entre elas: interpretação do sinal de radar, reconhecimento de caracteres, reconhecimento de fala, diagnósticos de radiografias, análise de células sangüíneas, entre outras [Sou] [Sep].

O estudo do reconhecimento de padrões apresenta três abordagens distintas: discriminante, estrutural e adaptativa [Sch].

Na abordagem discriminante, são extraídas características mensuráveis do padrão e o reconhecimento é feito através de um fator discriminante, que posiciona este padrão num espaço de características.

Na abordagem estrutural, um padrão é subdividido em vários sub-padrões mais simples, os quais também podem ser subdivididos diretamente e, assim, sucessivamente, até alcançarem primitivas suficientemente simples para serem reconhecidas. Esta abordagem é utilizada com padrões mais complexos, onde o número de características é muito grande.

Já o reconhecimento adaptativo requer a adição de processamento distribuído e paralelo aos métodos tradicionais de reconhecimento. Também conhecida como reconhecimento de padrões neural, esta abordagem tenta reproduzir a forma como os sistemas neurais biológicos armazenam e manipulam informações. Estes sistemas de reconhecimento utilizam as redes neurais artificiais (RNAs), que também são objeto de estudos desta dissertação.

O primeiro modelo de neurônio foi desenvolvido na década de 40 pelo neurofisiologista McCulloch e pelo matemático Walter Pitts, da Universidade de Illinois. Assim, as RNAs foram baseadas na analogia entre células nervosas vivas e um processo eletrônico envolvendo resistores variáveis e amplificadores.

A partir daí, muitos progressos foram feitos. A tolerância a falhas, a grande capacidade de memória e a capacidade de processamento em tempo real do sistema nervoso humano sugeriram uma arquitetura para as redes neurais artificiais. O tempo mínimo para o processamento de uma informação pelo sistema nervoso humano, lento em relação aos

dispositivos eletrônicos, sugeriu que a base da computação biológica fosse composta de alguns pequenos passos seriais, cada um deles processados paralelamente. Desta forma, as redes neurais artificiais seriam baseadas num conjunto de conceitos bastante conhecidos que vão desde o reconhecimento de padrões biológicos até os paradigmas da computação paralela.

Este trabalho enfoca a abordagem adaptativa para reconhecimento de padrões, utilizando-se as redes neurais artificiais no desenvolvimento de um sistema de reconhecimento de fala. Este sistema pode identificar comandos verbais ou palavras que significam tarefas as quais um robô irá executar de maneira autônoma. Esta representa a forma de interfaceamento mais natural para o operador humano.

Para identificação dos comandos falados pelo locutor utiliza-se como modelo acústico uma rede neural multicamadas *Perceptron* com algoritmo de treinamento *Backpropagation.* A rede neural artificial com a qual se obteve o melhor desempenho possui dezesseis neurônios na camada de entrada, sessenta na camada intermediária e quatro na camada de saída. Esse sistema é adaptativo a qualquer locutor desde que seja submetido a uma etapa de treinamento. Até o presente momento a rede foi treinada para reconhecer quatro comandos: "abre", "fecha", "liga" e "desliga".

No capítulo 2 serão apresentados os principais aspectos referentes ao reconhecimento de padrões, sendo o reconhecimento de fala uma de suas aplicações.

No capítulo 3 alguns conceitos a respeito das redes neurais serão apresentados.

No capítulo 4 serão descritos os procedimentos adotados no desenvolvimento do sistema de reconhecimento de fala.

No capítulo 5 os resultados obtidos da aplicação deste sistema a um robô são apresentados.

No capítulo 6, são colocadas, finalmente, as conclusões a respeito do trabalho desenvolvido e as perspectivas de futuros trabalhos neste contexto.

Capítulo 2

RECONHECIMENTO DE FALA

2.1. INTRODUÇÃO

O objetivo deste capítulo é fornecer os fundamentos básicos sobre reconhecimento de fala. Assim, serão apresentados os principais conceitos envolvidos na área de reconhecimento de fala e o relacionamento destes com a área de reconhecimento de padrões. Em seguida, serão descritos alguns fatores que influenciam na classificação destes sistemas e a composição funcional de cada módulo envolvido nos mesmos.

2.2. FUNDAMENTOS DE RECONHECIMENTO DE PADRÕES

Podemos situar o reconhecimento de fala como um subconjunto de uma área mais abrangente conhecida como reconhecimento de padrões. Assim, alguns conceitos importantes referentes a área de reconhecimento de padrões serão apresentados a seguir [Sch]:

• **Padrão:** Um padrão é um arranjo ou ordenação em que pode-se dizer que exista alguma organização estrutural. Um padrão pode ser referenciado como uma quantidade ou descrição estrutural de um objeto ou algum outro item de interesse, podendo ser complexo ou tão básico quanto um conjunto de medidas ou observações, como peso, idade, número de peças, por exemplo, geralmente sendo representado na forma de vetor ou matriz.

• **Características (*features*):** As características podem ser entendidas como qualquer medição útil extraída de um processo, podendo ser, por exemplo, o resultado da aplicação de um algoritmo ou operador de extração de características nos dados de entrada. Intensidade de sinais e descrições geométricas de uma região são exemplos de *características*. Um grande esforço computacional é requerido neste processo, podendo os dados, muitas vezes, conter erros ou 'ruídos'.

• **Vetor de características e espaço de características:** vetor de características é um vetor de dimensão d, denotado por x, que contém as características de forma organizada. Já um espaço de características é um espaço multidimensional que contém os vetores de características. Se, por exemplo, todas as características são números reais, pode-se dizer que o espaço de características é R^d.

• **Classificação:** a classificação consiste na atribuição de dados de entrada a uma ou mais classes *c* pré-especificadas. Esta atribuição é baseada na extração de características significativas ou atributos, e no processamento ou análise destes atributos.

• **Reconhecimento:** o reconhecimento consiste no processo de classificação. Quando são analisados problemas de reconhecimento de padrões, são utilizadas c + 1 classes, visto que uma classe deverá conter a saída *não classificado*, *não conhecido* ou *não pode decidir*.

• **Classe de padrões:** é um conjunto de padrões que possuem características similares. A questão em muitas aplicações de reconhecimento de padrões é identificar os atributos apropriados e formar uma boa medida de similaridade.

• **Pré-processamento:** é o processo de filtragem ou transformação dos dados de entrada para auxiliar na praticidade computacional, extração das características e minimização do ruído.

A partir destes conceitos, pode-se dizer que, dado um conjunto de características que definem um determinado objeto, a tarefa de reconhecer padrões consiste em fazer com que tais características sejam identificadas. De modo formal, o reconhecimento de padrões pode ser entendido como um mapeamento "opaco" entre o espaço padrão e o espaço da classe

associada [Zad]. O mapeamento é dito "opaco" porque a função que leva ao espaço da classe associada não é conhecida. Outra definição, dada por Fu em [Fu], descreve o reconhecimento de padrões como uma associação de um padrão a uma determinada classe.

Um objeto "x" pode ser representado por muitas instâncias ("x_n"), com aparências distintas umas das outras e, além disso, podem ser caracterizados como conceituais ou físicos, ou situações ou eventos. A Figura 2.1 ilustra o processo de reconhecimento de padrões, sendo $R_{op}(x)$ denominado mapeamento "opaco", que leva às classes de padrões C(x). Nos seres humanos, o processo de reconhecimento também ocorre de forma "opaca", através de seus componentes perceptivos e cognitivos.

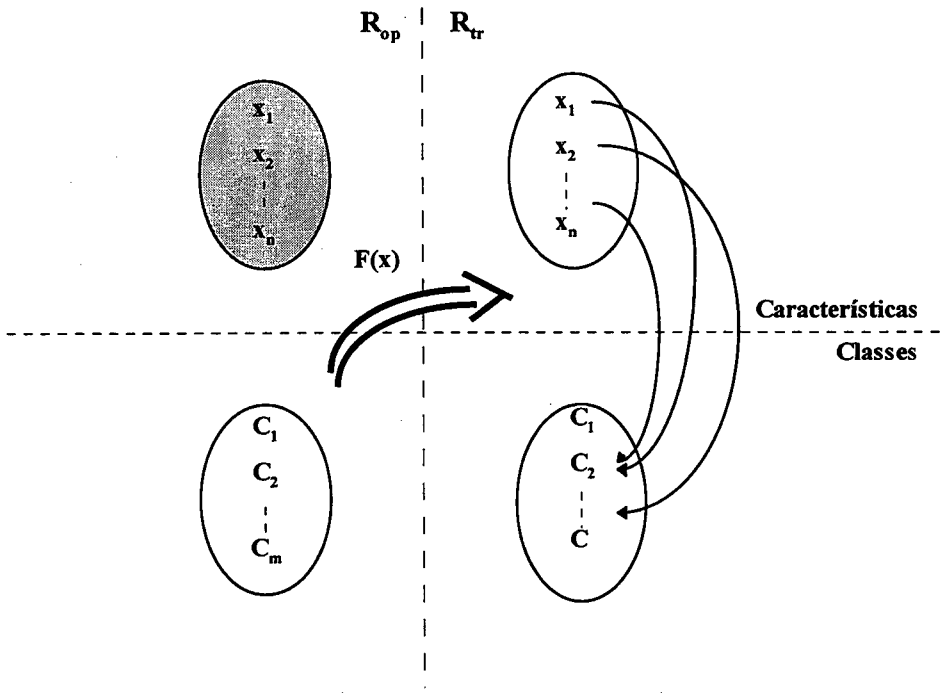

Figura 2.1. Transformação de um mapeamento opaco para um mapeamento transparente.

Para que o reconhecimento de padrões possa ser realizado automaticamente, é necessário que $R_{op}(x)$ se torne conhecido. Em outras palavras, procura-se transformar o mapeamento "opaco" num mapeamento "transparente". Para que isso seja possível, é necessário conhecer a transformação matemática F(x), que leva de R_{op} a R_{tr} (mapeamento "transparente"). Obtida F(x), através de cálculos matemáticos efetuados sistematicamente se poderá obter a relação $R_{tr}(F(x)) \Rightarrow C(x)$. Por outro lado, não há, geralmente, um princípio

básico para a escolha da transformação F(x). Ou seja, normalmente não se sabe quais aspectos de uma representação são apropriados para reconhecer os padrões desejados.

A escolha de R_{tr} é outro passo essencial para o reconhecimento computacional de padrões. Isto porque, apesar de possuirmos sensores de alta qualidade, capazes de quantificar características como temperatura, fragrância, textura, entre outras [Uhr], é necessário que se saiba distinguir quais características são necessárias em cada situação. Quais características contribuem, quais são redundantes e quais são irrelevantes para o reconhecimento? Esta é uma questão difícil de ser respondida, mas é uma tarefa essencial na implementação de um sistema de reconhecimento de padrões, denominada de pré-processamento. Na seção seguinte será apresentada uma classificação das técnicas de pré-processamento relevantes para o presente trabalho.

2.2.1. Classificação das técnicas de pré-processamento

• **Codificação e aproximação:** as técnicas de pré-processamento incluídas neste grupo prevêem a codificação dos padrões, ou seja, sua representação de forma digital, a fim de que possam ser processados por computador. Além de digitalizados, estes padrões são representados de forma compacta e algumas vezes aproximados, mas sem que a informação em questão seja degradada.

• **Filtragem, restauração e reforço:** neste caso, operações invariantes no tempo e invariantes no espaço, como a Transformada de Fourier, Hadamar e Wavelet, por exemplo, podem ser utilizadas para detectar um dado padrão (filtragem), restaurá-lo e reforçá-lo.

2.2.2. Formatos de representação dos padrões

• **Representação numérica e não-numérica:** em geral, um padrão é uma estrutura de características que inclui informações a respeito do nome e dos valores das mesmas, além de informações explícitas ou implícitas sobre as relações entre elas, se existirem. Não existe

uma classificação de formatos de padrões; entretanto, baseados na natureza de suas características, podemos citar as representações numérica e não-numérica.

Exemplo 1: Representação numérica

Codificação das informações do tempo:

tempo ≡ (temperatura em °C: 18.2, altura barométrica em mmHg: 759.8, umidade relativa em percentual: 70)

≡ (18.2, 759.8, 70)

Exemplo 2: Representação não-numérica

Codificação das informações de uma maçã:

maçã ≡ (categoria: fruta, cor: vermelha, sabor: doce)

≡ (fruta, vermelha, doce)

2.2.3. Etapas na construção de um sistema de reconhecimento de padrões adaptativo

Geralmente, são três as etapas identificadas na construção de um sistema de reconhecimento de padrões. São elas:

- **Treinamento**: neste estágio, estima-se um conjunto de parâmetros do modelo através de um método denominado critério de treinamento. Isto é feito sistematicamente, até que o erro de estimação global seja mínimo, de tal forma que o modelo "aprenda" a correspondência entre as características dos objetos e seus respectivos rótulos.

• **Teste**: neste segundo estágio, aplica-se ao modelo um conjunto de dados - características e rótulos - diferentes dos dados usados na etapa de treinamento, a fim de verificar o desempenho geral do sistema. Estes dados são chamados "validação cruzada" (*cross-validation).*

• **Implementação**: este é o estágio final, onde as características que possuíam rótulos desconhecidos, após serem submetidas ao sistema, passam a ter na saída rótulos conhecidos.

Do ponto de vista estrutural, um sistema de reconhecimento de padrões consiste de um **extrator de características** e um **classificador** (Figura 2.2). O extrator de características tem por finalidade normalizar os dados coletados e transformá-los para o espaço de características. No espaço de características, os dados são representados e comprimidos de tal forma que objetos pertencentes a mesma classe sejam semelhantes em algum sentido, e que exista uma clara distinção entre objetos de outras classes.

O classificador, por sua vez, tem como função receber as características evidenciadas pelo extrator e identificar a que classe estas características pertencem. Para isso, pode-se utilizar a técnica de padrões de referência (*templates*) (item 2.3.2.5) ou o cálculo probabilístico (*likelihood*) (item 2.3.2.6). O classificador deve, *a priori*, passar por um estágio de treinamento, a fim de que seja estabelecida uma relação entre as características e o rótulo de uma determinada classe.

A estrutura para um sistema de reconhecimento de padrões, ilustrada na Figura 2.2, tem-se mostrado eficiente, considerando-se que as condições de treinamento e teste sejam comparáveis [Skl-a]. Porém, tais condições são, na realidade, bastante suscetíveis a distorções.

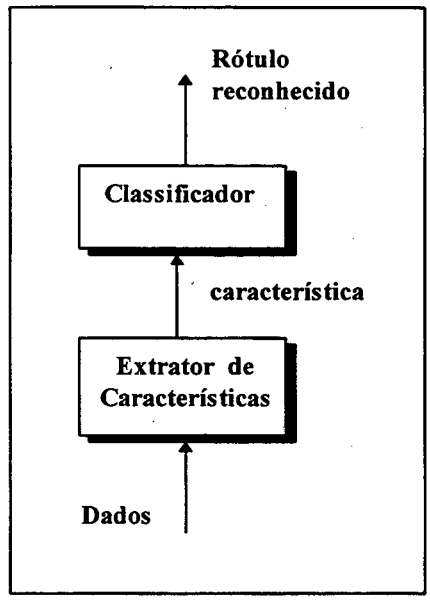

Figura 2.2. Estrutura de um sistema de reconhecimento de padrões.

2.3. A FALA COMO UM PADRÃO

A fala é o meio de comunicação mais utilizado pelos seres humanos, é o meio mais natural e confortável de interação. Portanto, imagina-se que assim também seja com relação à interação entre os homens e as máquinas.

Duas dificuldades são encontradas para tornar esta suposição uma realidade. A primeira delas é a alta não-linearidade do processo de produção da fala, influenciado por vários fatores, como idade, sexo e estado emocional e de saúde do locutor. A segunda, é o alto grau de variação na percepção da fala, também influenciada por ruídos de fundo, ambientes acústicos e características do meio de transmissão [Fan].

Neste sentido, grandes progressos têm sido feitos a partir de 1970. Porém, ainda não existem, atualmente, sistemas capazes de realizar a complexa tarefa de reconhecimento de fala tão bem quanto o ser humano, a não ser sob de determinadas condições de avaliação [You].

2.3.1. Precisão

As condições de avaliação e de precisão podem variar de acordo com as seguintes dimensões:

- Tamanho do vocabulário: **atualmente ainda não existe nenhuma definição estabelecida, embora tente-se seguir a da Tabela 2.1.**

Número de palavras	Tamanho do vocabulário
10	Pequeno
100	Médio
1000	Grande
10000	Muito grande

Tabela 2.1. Tabela de definição de tamanhos de vocabulários.

- **Independência *versus* dependência de locutor**: um sistema dependente de locutor é o que reconhece a voz de uma única pessoa ou vozes que tenham características semelhantes a esta. Estes sistemas de reconhecimento de fala são mais fáceis de desenvolver, têm um custo financeiro menor e têm maiores possibilidades de alcançar precisão. Um sistema de reconhecimento de fala que trabalha com mais flexibilidade em relação ao locutor é dito ser independente. Nestes, o custo é maior, além de serem mais difíceis de desenvolver e de alcançar uma precisão satisfatória. Outra classe de sistemas de reconhecimento de fala é a adaptativa ao locutor. Nestes, o sistema se adapta a pessoas ou locutores com novas características.

- **Palavras isoladas e fala contínua**: é necessário que as palavras sejam separadas por pausas ou silêncios artificiais para que sejam reconhecidas por sistemas que trabalham com palavras isoladas. Isto não é exigido nos sistemas que trabalham com fala contínua. Estes são mais difíceis de implementar, pois não existe pausa artificial para caracterizar o início e fim de uma palavra. Assim, a pronúncia de uma palavra pode interferir na palavra subseqüente; este efeito é chamado co-articulação.

- **Limitação de tarefas e/ou linguagem**: mesmo com um vocabulário fixo, existe um grande número de possibilidades para a formação de uma sentença a partir deste

vocabulário. Portanto, devem ser estabelecidos limites para as tarefas e para a linguagem (semântica e/ou sintática), de forma a auxiliar o sistema de reconhecimento. Estes limites são estabelecidos por gramáticas.

• **Condições adversas**: um sistema também pode ser degradado por uma série de fatores, tais como: ruído ambiente, distorções acústicas, diferentes microfones, largura de banda e freqüência limitada, e maneiras alteradas de falar.

2.3.2. Composição e Funcionalidade

2.3.2.1. Estrutura geral de um sistema de reconhecimento de fala

A estrutura padrão de um sistema de reconhecimento de fala é dividida em módulos, cada um com uma função específica (Figura 2.3) [Fan] [Lip-b] [Lip-c] [Koh-b] [Wai-a]. A seguir, será dada uma descrição abstrata desta estrutura para, posteriormente, ser feita uma descrição mais detalhada de cada módulo.

A entrada para o sistema é uma forma de onda amostrada do sinal de áudio, capturada por microfone (Figura 2.3). O próximo estágio é o de pré-processamento. Neste estágio, também chamado de extrator de características ou *front-end*, é computada uma seqüência de características, na maioria das vezes derivada das representações espectrais ou cepstrais (item 2.3.2.3) da fala. O estágio de modelamento acústico é responsável por modelar um conjunto de sons de fala, obtidos em uma dada observação, em determinada unidade de tempo. Para cada unidade atômica de sons modelada, este estágio fornece, por exemplo, uma probabilidade de ocorrência local. Com isso, será gerado o quadro de pontos que será utilizado em um estágio de pesquisa (decodificador ou *decoder*) para identificação da palavra. Para minimizar a taxa de erros, podem ser fornecidas ao decodificador, antecipadamente a estimação do modelo, informações adicionais sobre as probabilidades de uma seqüência de palavras. Essas informações adicionais são conhecidas como modelo de linguagem ou gramática.

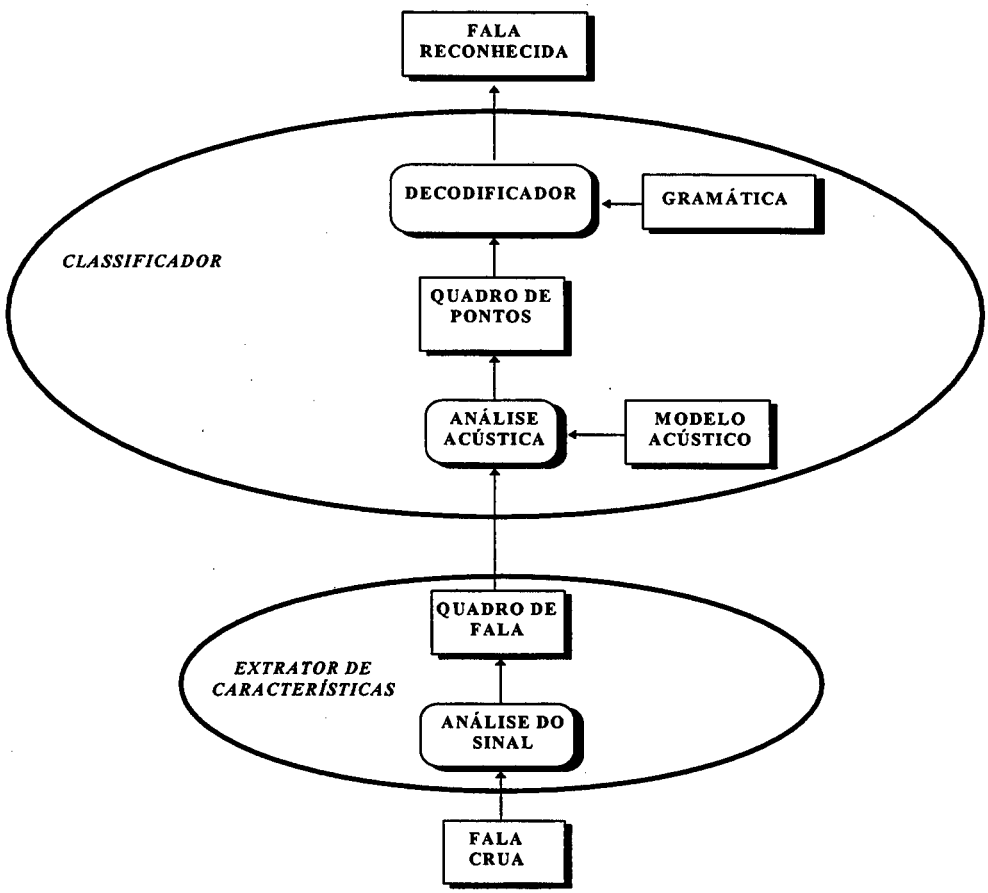

Figura 2.3. Estrutura de um sistema de reconhecimento de fala.

2.3.2.2. Aquisição do sinal de fala

A fala é tipicamente amostrada em altas freqüências: 16 KHz sobre um microfone ou 8 KHz sobre um telefone, por exemplo [Koh-b]. É importante lembrar que as características ambientais, a espécie de microfone e o transdutor A/D que são utilizados para gravar o sinal de áudio podem ter diferentes efeitos sobre a representação e o reconhecimento da fala. Recentemente, grandes esforços têm sido feitos no desenvolvimento dos chamados sistemas robustos, os quais toleram diferentes espécies de microfones, características ambientais e condições de ruído. A Figura 2.4 ilustra dois exemplos de sinais de fala para as palavras "abre" e "fecha".

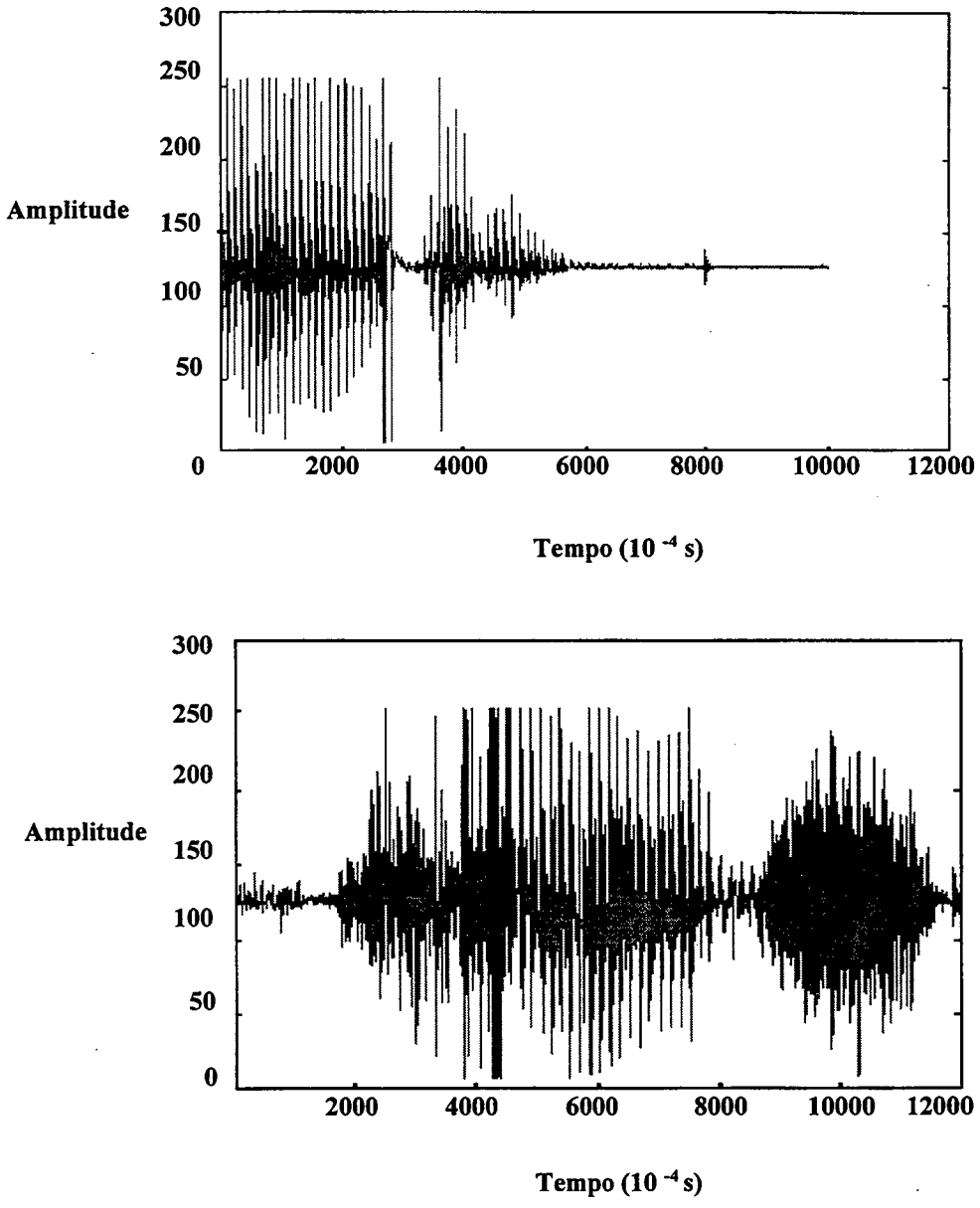

Figura 2.4. Sinais amostrados das palavras ABRE e FECHA, respectivamente

2.3.2.3. Análise do sinal

A fala em seu formato original pode ser inicialmente transformada e/ou comprimida com a intenção de simplificar o processamento subseqüente. Existem muitas técnicas de análise de sinal, com as quais podem se extrair características úteis e comprimir dados sem que haja perda de informações. Dentre as técnicas mais conhecidas, podemos citar [Lip-c] [Sou]:

• **Análise de Fourier (FFT[1]):** produz freqüências discretas sobre o tempo e tem como vantagem a interpretação gráfica. Essas freqüências geralmente são distribuídas em uma escala linear e logarítmica, para baixas e altas freqüências, respectivamente, correspondendo às características fisiológicas do ouvido humano [Coo]. A Figura 2.5 apresenta a FFT para os sinais de fala apresentados anteriormente na Figura 2.4.

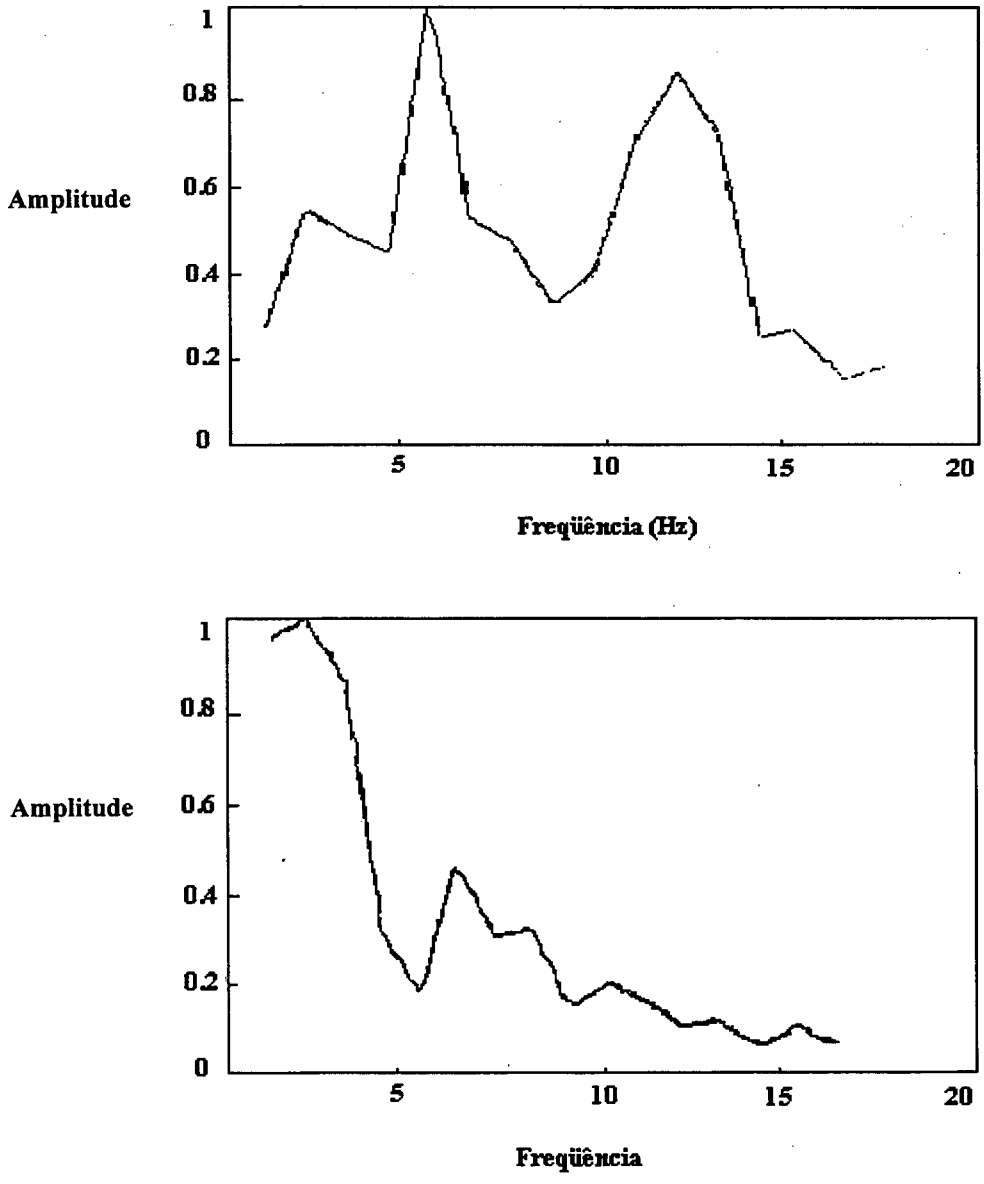

Figura 2.5. FFT normalizado dos sinais que representam as palavras ABRE e FECHA, respectivamente.

[1] Fast Fourier Transform

• **Análise Cepstral:** os coeficientes são obtidos pelo cálculo da transformada inversa logarítmica do espectro de potência do sinal. O sinal de áudio original, composto pela superposição da fala e do ruído é representado, na freqüência, pela multiplicação do sinal de fala x pelo ruído y, x.y. Após aplicarmos o logaritmo, teremos log x + log y podendo-se, assim, separar e remover a componente de ruído. A seguir, através da transformada inversa de Fourier obtém-se o sinal de fala [Wai-b].

• **Análise de Wavelet:** na Transformada de Fourier, um sinal x(t) é representado na freqüência como a soma ponderada das funções periódicas escolhidas como base para a transformação seno e cosseno. Já na Transformada de Wavelet, as funções escolhidas como base para a transformação são as chamadas Wavelet's mãe, que têm como característica a não periodicidade e, conseqüentemente, a localização temporal. Esta característica proporciona, além de informações do espectro, a informação adicional sobre o momento no tempo em que os coeficientes espectrais ocorrem. O gráfico que representa o sinal transformado tem agora três dimensões, sendo a terceira o eixo do tempo [Hay-a] [Wis].

2.3.2.4. Quadro de fala

Como conseqüência da etapa de análise do sinal, tem-se uma seqüência de quadros. Estes, geralmente, possuem duração de 10 ms, com cerca de 16 coeficientes por quadro, que serão posteriormente usados para análise do modelo acústico. Esses valores são tomados partindo-se do princípio de que nesta faixa o sinal de fala tem um comportamento estacionário [Fan] [Koh-b] [Pra].

2.3.2.5. Modelo acústico

A fim de analisar os quadros de fala, nós necessitamos de um modelo acústico. Existem muitos tipos de modelos acústicos, que diferem entre si por suas propriedades, além do modo como são representados. Entre as representações mais conhecidas, temos:

• *Templates*: é a forma de representação mais simples, na qual apenas uma amostra da unidade de fala a ser modelada é armazenada. Uma palavra ou fonema pode ser

identificada através de sua simples comparação com *"templates"* conhecidas. Neste tipo de representação, o principal problema é que, como a variação acústica é muito grande, torna-se difícil modelar as unidades de fala, já que seriam necessárias múltiplas *templates* para cada unidade.

- *Estados*: uma representação mais flexível do que *templates*, é baseada no treinamento do modelo acústico, através da utilização de estados. Neste tipo de representação, cada palavra é modelada por uma seqüência de estados treinados. Cada estado indica o som que provavelmente pode ser ouvido naquele segmento de palavra, usando uma distribuição probabilística sobre o espaço acústico. As distribuições probabilísticas podem ser modeladas de forma paramétrica ou não-paramétrica. No caso da modelização paramétrica, pode-se utilizar, por exemplo, uma distribuição *gaussiana*; no caso da modelização não-paramétrica pode-se utilizar, por exemplo, um histograma ou uma rede neural artificial, que é a proposta deste trabalho.

2.3.2.6. Análise acústica e quadros de pontos

A análise acústica consiste na comparação de cada quadro de fala com o modelo acústico. Este processo gera um quadro denominado quadro de pontos, que indica o grau de semelhança entre o quadro de fala e o modelo. Para modelos baseados em *templates*, um quadro de pontos é construído a partir da distância euclidiana entre o *template* e um quadro desconhecido. Já para o modelo baseado em estados, um quadro de pontos é composto de probabilidades de emissão, isto é, o *likelihood* do quadro de fala gerar o estado atual, o que é determinado pela função de estados.

2.3.2.7. Gramáticas

A fim de minimizar a taxa de erro do sistema de reconhecimento de fala, pode-se utilizar uma gramática. Existem gramáticas a nível de tarefas e a nível de linguagem; sendo que, a nível de linguagem, pode ser realizada uma análise semântica ou sintática.

No caso de uma gramática a nível de tarefas, o sistema deverá reconhecer um conjunto limitado das mesmas. Por exemplo: se um sistema de auxílio a lista telefônica estiver apto a fornecer o endereço e o número telefônico de determinado assinante, ele será incapaz de fornecer, por exemplo, o saldo bancário do mesmo, já que esta tarefa não consta de seu conjunto de tarefas.

No caso de uma gramática a nível de linguagem semântica existem restrições em relação às palavras pronunciadas. Por exemplo: se o sistema está apto a reconhecer a palavra "necessito" em uma determinada frase, esta não poderá ser substituída por um sinônimo, como "preciso". Já uma gramática a nível de linguagem sintática, impõe restrições em relação a ordem das palavras que compõem a frase. Por exemplo: se o sistema está apto a reconhecer a frase "tenha uma boa noite", ele não reconhecerá a frase "tenha uma noite boa".

2.4. SUMÁRIO

Neste capítulo foram apresentados conceitos fundamentais sobre reconhecimento de fala. A estrutura dos sistemas de reconhecimento de fala foi abordada, apresentando-se, primeiramente, uma visão abstrata do problema para, em seguida, detalhar-se cada módulo que os compõem. Além disso, alguns dos problemas envolvidos no processo de reconhecimento, tais como: não-linearidade do sinal de fala, condições acústicas variantes e a instabilidade do sinal de voz de cada locutor, foram citadas.

No capítulo seguinte serão apresentados alguns fundamentos das redes neurais artificiais, citando-se arquiteturas, tipos de treinamento e principais propriedades de algumas redes neurais relevantes.

Capítulo 3

REDES NEURAIS ARTIFICIAIS

3.1. INTRODUÇÃO

No presente capítulo nós iremos rever as origens do cognitivismo e a teoria na qual está fundamentado. Além disso, veremos alguns conceitos, a composição básica de um neurônio, arquiteturas de RNAs e tipos de treinamento. Em seguida, alguns modelos de redes neurais artificiais relevantes para o reconhecimento de fala serão citados.

3.2. HISTÓRICO

Há muitos anos, o ser humano vem tentando conceber artefatos capazes de ter um comportamento inteligente. Segundo Barreto, em [Bar], algumas experiências foram feitas neste sentido, mas nenhuma delas reconhecida, até que Ada Lovelace, em 1842, comentou sobre a máquina de Babbage como sendo um artefato possuidor destas potencialidades. Posteriormente, Leonardo Torres y Quevedo, em 1890, construiu uma máquina, hoje chamada autômata, para terminar uma partida de xadrez (rei e torre contra rei). Outras contribuições foram dadas por Turing, que propôs um teste para verificar se uma máquina possui inteligência.

Paralelamente, a partir de 1940, McCulloch e Pitts mostraram que um neurônio pode ser modelado como um dispositivo limitador, a fim de realizar funções lógicas.

Em 1956, nos Estados Unidos, aconteceu o primeiro encontro entre os pesquisadores mais importantes há duas décadas: McCarthy, Minsky, Newell e Simon, objetivando o estudo do comportamento inteligente. A partir deste encontro e do livro "Automata Studies", surgem, segundo Barreto em [Bar], o primeiro artigo de RNA como sendo um paradigma da arquitetura computacional e duas abordagens para o estudo do comportamento inteligente: simbólica (ou globalista) que reproduz o comportamento inteligente não levando em consideração a maneira como a inteligência humana foi criada, e conexionista (ou reducionista), que preocupa-se em criar estruturas semelhantes às estruturas biológicas neurais, a partir das quais surge o comportamento inteligente.

As redes neurais artificiais têm suas pesquisas continuadas por Rosenblatt (Perceptron), Widrow (Adaline), e Steinbuck (Matriz de aprendizado) que, em 1960, apresentaram um modelo de neurônio mais refinado. O Perceptron, quando foi introduzido, teve consideráveis atenções devido a sua simplicidade conceitual. Entretanto, em 1969, Minsky e Papert provaram que o Perceptron não podia ser usado para funções lógicas complexas. Portanto, o interesse pelas redes neurais artificiais só foi revigorado a partir das contribuições de Hopfield que, além de apresentar um modelo consistindo de equações diferenciais (não-lineares) de primeira ordem (máquina de Boltzmann) que minimiza uma certa função energia (1982), também argumentou que existiam potencialidades computacionais extremamente maiores a nível de redes neurais do que a nível de neurônio. Em 1986, Rumelhart, publicou um algoritmo de aprendizado denominado Backpropagation, que pode treinar uma rede multicamadas e realizar tarefas nas quais o Perceptron falhava. Assim, a partir do Backpropagation, as redes neurais artificiais ganharam maior atenção por parte da comunidade científica.

Atualmente, pesquisadores de várias áreas como psicólogos, engenheiros e matemáticos tentam entender melhor o funcionamento do cérebro para dar continuidade à evolução das RNAs. Segundo Karayiannis em [Kar], as tendências das pesquisas nesta área apontam para: o desenvolvimento de algoritmos de aprendizagem mais rápidos, incluindo questões sobre convergência e estabilidade; desenvolvimento de arquiteturas altamente modulares e interconectadas; o desenvolvimento de técnicas de implementação (eletrônica, óptica e biológica) e o desenvolvimento de aplicações.

3.3. FUNDAMENTOS BIOLÓGICOS

A criação de Redes Neurais Artificiais foi baseada nos neurônios biológicos (Figura 3.1) e nos sistemas nervosos. Camilo Golgi, em 1875, deu um dos primeiros passos na neuroanatomia com a descoberta de um método que possibilita isolar e observar de forma individual os neurônios através do uso de corantes [Bar]. A partir do método de Golgi, Cajal [Caj] apresentou dois resultados muito importantes: o primeiro foi a adoção da idéia inicial de sistema nervoso, que postulava que a comunicação entre as células era realizada através da sinapse[2], (antes disso, através de Golgi, sabia-se apenas que existia um número grande de células isoladas sem sugerir que haviam interligações entre elas com o sentido de formar uma rede). O segundo foi de que a interconexão entre neurônios não se dava ao acaso, mas sim de maneira altamente específica e estruturada.

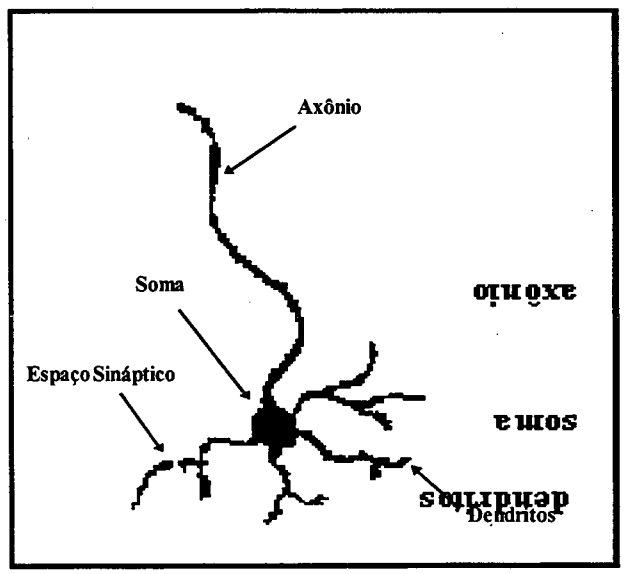

Figura 3.1. Neurônio biológico

Atualmente, sabe-se que o neurônio tem um corpo celular denominado soma e várias ramificações denominadas dendritos que são responsáveis por conduzir os sinais elétricos

[2] Ponto de conexão entre neurônios.

das extremidades para o corpo celular dos neurônios. As ramificações que tem por função transmitir o sinal do corpo celular às extremidades são denominadas axônios e, geralmente, são uma só ramificação. O axônio é conectado aos dendritos de outros neurônios ou a outros axônios [Bar].

De acordo com James Weston, descobridor do DNA, o cérebro é a peça mais complexa do maquinário biológico existente na Terra. Com o intuito de entender as funções do cérebro, neurobiologistas estudaram curvas características do estímulo-resposta de neurônios e, juntamente, redes de neurônios; paralelamente, psicólogos estudavam tais funções a nível comportamental e cognitivo. Ao longo dos anos, observou-se que estas duas abordagens vêm convergindo em seus estudos, porém, ainda serão necessários muitos anos de pesquisa para o completo entendimento deste órgão de aproximadamente 10^{11} neurônios,que recebe e envia sinais elétricos através de até 10 mil sinapses, por neurônio [Bar].

3.4. REDES NEURAIS ARTIFICIAIS

As redes neurais são inspiradas no modelo de processamento do cérebro biológico. Desta forma, podemos caracterizar uma rede neural artificial como um grande número de elementos simples de processamento (neurônios) os quais influenciam-se uns aos outros de forma excitatória ou inibitória. Assim, cada unidade, considerando a função de transferência, calcula uma soma de pesos não lineares de suas entradas e difunde o resultado sobre suas conexões de saída a outras unidades.

Esta estrutura executa milhões desses pequenos passos seqüenciais paralelamente e, desta forma, tenta emular o cérebro biológico. Quando faz o ajuste dos pesos, a rede neural se adapta de forma a produzir um conjunto de saídas consistente a partir de um determinado conjunto de entradas. Por esse comportamento, diz-se que as redes podem "aprender".

3.5. PROPRIEDADES DAS REDES NEURAIS ARTIFICIAIS

A seguir serão citadas as principais propriedades que motivam a utilização das redes neurais artificiais na aplicação de sistemas de reconhecimento de fala.

• **Paralelismo**: as redes são altamente paralelas por natureza, assim, é recomendada sua implementação sobre computadores paralelos, permitindo maior rapidez no processamento dos dados.

•**Capacidade de treinamento:** as redes pode ser treinadas para se adequar a qualquer padrão de entrada e saída. Isto pode ser usado, por exemplo, para uma rede aprender a classificar padrões de fala.

• **Generalização:** as redes não memorizam somente os dados treinados. Elas podem, a partir de dados treinados, generalizar seu procedimento para novos padrões de entrada. Isto é essencial para o caso de sistemas de reconhecimento de fala, porque os padrões acústicos nunca são exatamente os mesmos.

• **Não linearidade:** as redes podem representar não linearidades, funções paramétricas de suas entradas, habilitando-as a desempenhar arbitrariamente transformações complexas de dados. Isto é útil, já que a voz é um processo altamente não linear.

• **Robustez:** as redes são tolerantes a ambos os problemas: danos físicos e dados ruidosos; no caso de dados ruidosos as redes podem ajudar a formar melhores generalizações. Esta é uma valiosa característica, porque o padrão de voz é notavelmente ruidoso.

3.6. PARÂMETROS QUE DEFINEM UM MODELO DE RNA

Existem muitos tipos de modelos conexionistas, com diferentes arquiteturas, procedimentos de treinamento e aplicações, mas eles são todos baseados em alguns princípios comuns. A partir de alguns conceitos básicos aplicados de maneiras diversificadas pode-se obter diferentes tipos de redes que podem aprender a calcular funções implícitas, ou agrupar automaticamente dados de entrada, ou, ainda, gerar apresentações compactas de

dados, entre outras tarefas. Assim sendo, em seguida serão comentados alguns destes princípios.

3.6.1. Neurônio Artificial

Quando se fala em neurônio artificial (Figura 3.2), também denominado unidade, nodo ou elemento de processamento, imagina-se que este seja o mapeamento, no sentido mais complexo, dos neurônios biológicos. Na maioria das vezes, a relação entre eles é representada por um elemento de processamento simples. A computação é totalmente feita por estes elementos, não existe nada além disso, que tenha o compromisso de supervisionar e coordenar as atividades realizadas em uma rede neural artificial.

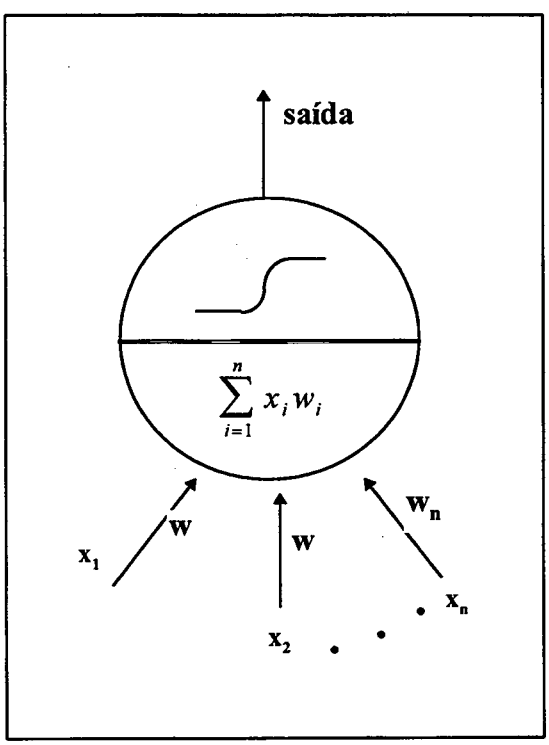

Figura 3.2. Neurônio artificial

Um neurônio é composto de várias entradas e uma única saída. A entrada de um neurônio recebe informações das saídas de outros neurônios e cada conexão entre estes recebe um valor denominado peso sináptico (*weight*). Assim, de acordo com a teoria expressa pela lei de Hebb [Fre] [Kar], quando um axônio de uma célula A está bastante

próximo para excitar uma célula B, e ocorrem mudanças repetida e persistentemente entre A e B, pode-se afirmar que há o aumento ou a diminuição da relação estabelecida por ambas as células. Ou seja, esta influência será representada pelos pesos de forma positiva (excitando) ou negativa (inibindo).

A representação dos pesos é feita através de um vetor de pesos (w_1, w_2, ... w_n). No caso mais geral, havendo mais de um neurônio na formação de uma rede, teremos, então, uma coleção de vetores representados matricialmente por W. Da mesma forma, todas as entradas de um neurônio são compostas por um vetor (x_1, x_2, ..., x_n), que serão utilizadas no cálculo do valor de ativação, também chamado função ativação ou simplesmente ativação.

Após o cálculo da ativação que é realizado pela multiplicação de todas as entradas de um neurônio por seus respectivos pesos (Equação 3.1), esta é comparada com um valor previamente estabelecido, chamado peso especial, cuja função é servir de limite para a ativação calculada. Se a ativação for superior a este limite, ela é aplicada em uma função de transferência e passada adiante através da saída. Caso contrário, o sinal representado pela ativação não é transferido para a unidade seguinte (Equação 3.2). Em ambos os casos, com a presença ou ausência do sinal, a resposta é significativa, pois afetará diretamente os neurônios posteriores ou a resposta final da rede.

$$\mathbf{a_i} = \sum_{i=0}^{n} \mathbf{x_i}\,\mathbf{W_i} \qquad \text{(Equação 3.1)}$$

$$\hat{y}_i = f(a_i) \qquad \text{(Equação 3.2)}$$

3.6.2. Função de Transferência

Uma observação a ser feita é que existem duas funções diferentes: a função ativação e a função de transferência, que muitas vezes são consideradas a mesma. Nos processamentos realizados pelos neurônios, a função de ativação antecede a função de transferência. Ela é de ordem interna e sua atribuição é decidir o que deve ser feito com a somatória das entradas ponderadas. Tomada esta decisão, a função de transferência tem por atribuição tomar este valor e produzir a saída do neurônio.

A função de transferência pode ter muitas formas como ilustrado na Figura 3.3, dependendo da aplicação, podendo ser simples ou complexa. Ela também é conhecida como um limiar lógico (*threshold*) e é responsável por definir e enviar para fora do neurônio o valor calculado pela função de ativação.

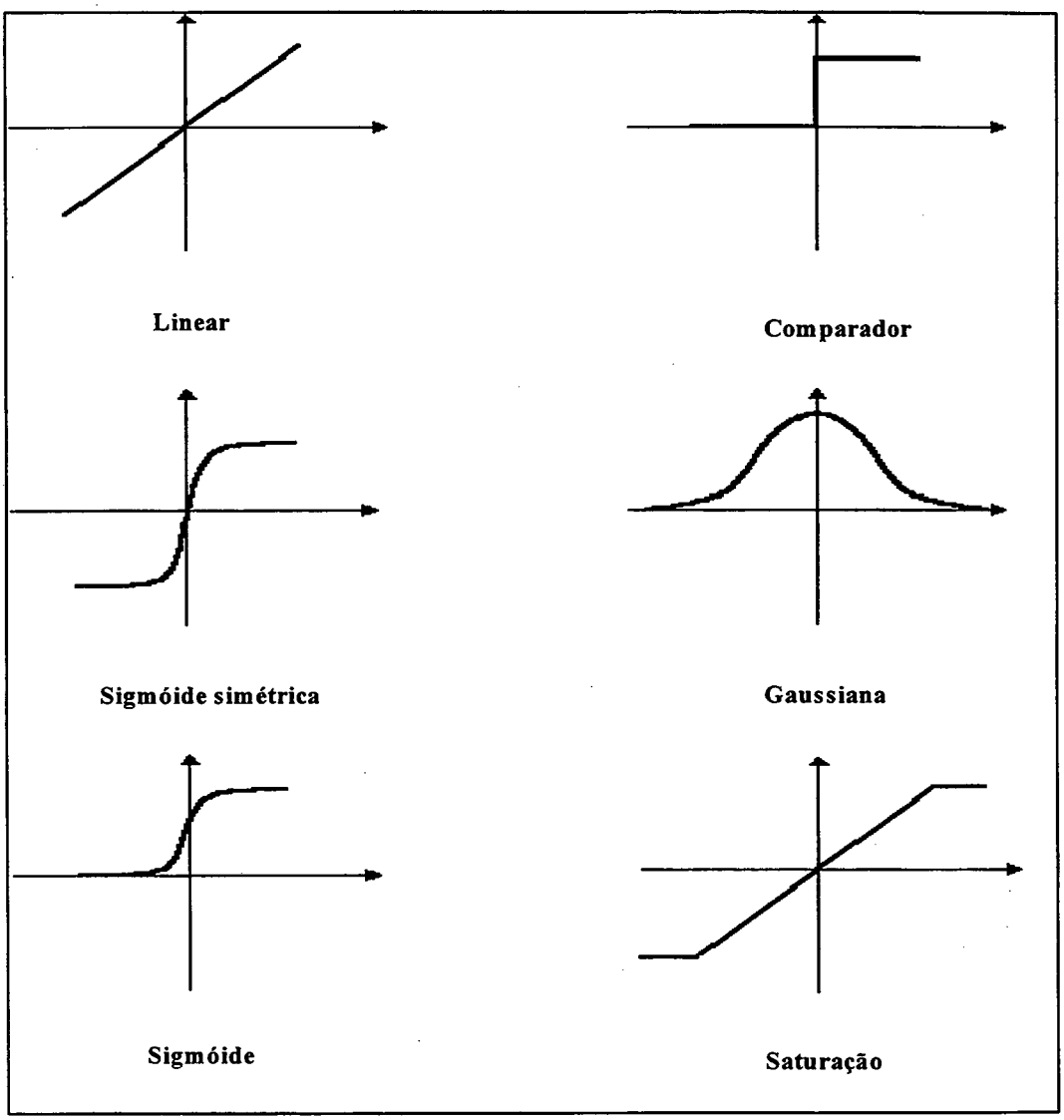

Figura 3.3. Funções de transferência

3.6.3. Treinamento

O treinamento de uma rede neural artificial consiste em submetê-la a um método sistemático. Este procedimento tem como objetivo providenciar que as conexões da rede

sejam adaptadas, de tal forma que esta rede possa exibir o comportamento computacional desejado para todas as entradas. Diz-se, então, que após aplicado este método a rede "aprendeu". Dentre os principais métodos sistemáticos utilizados para realizar o treinamento temos: Widrow-Hoff ou regra delta [Wid-a], regra delta generalizada ou Backpropagation [Rum-a], lei de Hebb [Fre], Conterpropagation [Fre] [Sep], entre outros.

Existem dois tipos de treinamento:

- **Supervisionado:** neste tipo de treinamento a rede utiliza pares de entrada e saída denominados conjunto de treinamento, o que significa que para cada entrada é associada uma saída desejada. Assim, toda vez que uma entrada for apresentada à rede, deverá ser verificado se a saída obtida confere com a saída desejada para aquela entrada. Se for diferente, a rede deverá ajustar os pesos até armazenar o conhecimento desejado. Este processo deverá ser repetido com todo o conjunto de treinamento (entrada e saída), até que a taxa de erro estipulada seja alcançada.

- **Não supervisionado:** Neste caso não existe saída desejada, sendo usados apenas os valores de entrada. A rede utiliza-se de medidas de correlação entre as entradas, identificando certas características de forma a classificá-las. Este tipo de treinamento é também conhecido como *competitivo* ou *auto-organizável.*

3.6.4. Redes simples camada e redes multi-camadas

Uma camada de uma rede neural artificial é composta de neurônios organizados de forma paralela entre si. Assim, uma rede é dita **simples camada** (Figura 3.4) se existe somente a camada de entrada, responsável pela coleta de dados para a rede, e a camada de saída que possui os neurônios necessários para a computação da rede e posteriormente enviam os resultados obtidos para o meio externo.

Nas redes **multi-camadas** (Figura 3.5) existem camadas denominadas intermediárias ou ocultas (*hidden*), que situam-se entre as camadas de entrada e saída. Neste tipo de rede, pode existir um número de camadas intermediárias ilimitado, possibilitando a extração de características de ordem superior, ou seja, o cálculo de funções não-lineares. Quando um

sinal é aplicado à rede, um vetor de entrada é construído a partir das unidades na camada de entrada. Este sinal será passado à primeira camada intermediária, por exemplo. Os sinais de saída desta camada intermediária servirão de entrada para camadas intermediárias posteriores, até que seja alcançada a camada de saída. O sinal de saída referente a esta camada constitui a resposta global da rede para o padrão de ativação fornecido nas unidades de entrada. Os procedimentos descritos neste item caracterizam o que chamamos estado da rede.

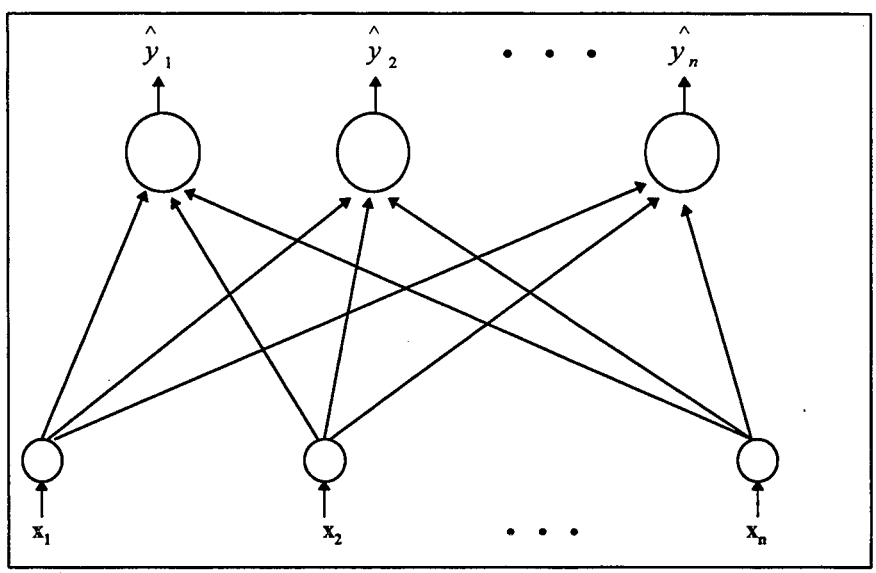

Figura 3.4. Rede simples camada

Uma observação a ser destacada é que qualquer rede multi-camadas com mais de uma camada intermediária pode ser equivalente a uma rede multi-camadas com somente uma camada intermediária [Rum-a].

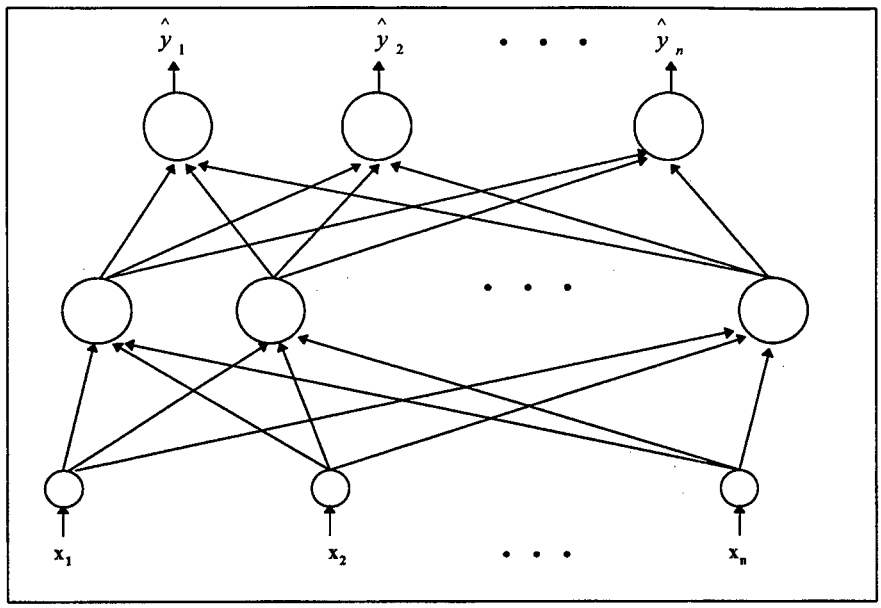

Figura 3.5. Rede multicamadas

3.6.5. Arquiteturas

Esta seção apresenta as arquiteturas de redes neurais mais utilizadas: *Feedforward* e *Feedback.*

3.6.5.1. Redes *Feedforward*

A característica de uma rede *feedforward* é não haver nenhuma forma de ligação da saída de neurônios pertencentes a uma determinada camada com neurônios que compõem a camada anterior a esta. Diz-se, portanto, não haver realimentação (*feedback)* (Figura 3.5). As arquiteturas de redes neurais *feedforward* que surgiram primeiramente foram: o Perceptron [Ros], e o Adaline [Wid-a]. Mas, somente a partir do aparecimento das redes *feedforward* multi-camadas [Rum-a] [Wid-b] [Lip-a] é que mostrou-se o potencial destas redes.

3.6.5.2. Redes *Feedback*

A existência de realimentação (Figura 3.6) proporciona à estas redes exibir um comportamento temporal. Desta forma, permite-se que os padrões analisados tenham aspectos espaciais e/ou temporais. Para o caso dos padrões analisados possuírem características espaço-temporais existe uma correlação temporal entre a seqüência de padrões. Um caso particular destas redes é encontrado quando a matriz de pesos é simétrica. O modelo de Hopfield é o mais simples e o mais utilizado neste tipo de arquitetura [Hop].

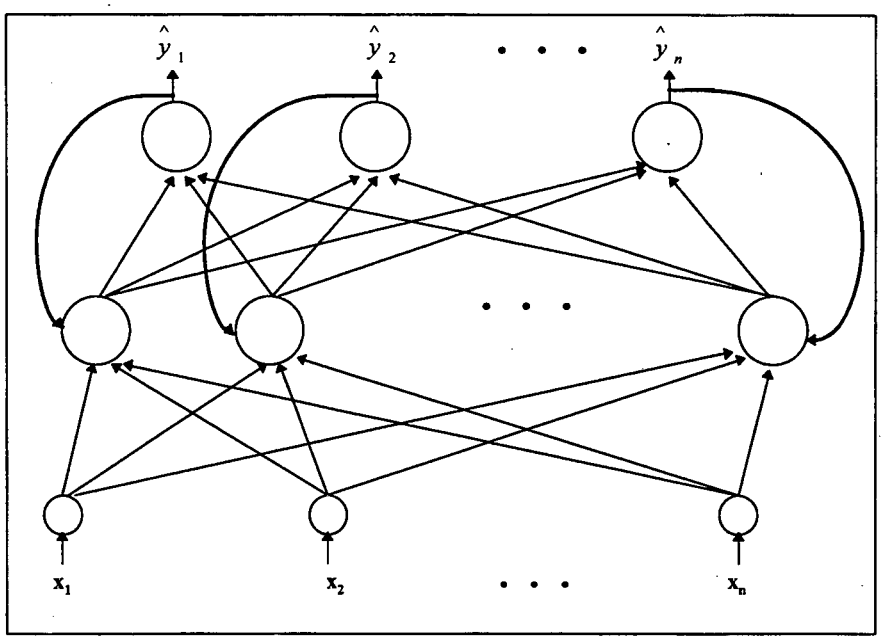

Figura 3.6. Exemplo de uma rede neural com realimentação

3.7. REDES NEURAIS RELEVANTES AO RECONHECIMENTO DE FALA

Nesta seção serão apresentados alguns exemplos de redes neurais artificiais que de alguma forma são relevantes para o reconhecimento de fala. Os atributos que as fazem relevantes não são muito claros, portanto, as redes citadas foram escolhidas por terem sido utilizadas por outros autores, como Lippmann em [Lip-b] [Lip-c], Hopfield em [Hop], Prager, Harrison e Fallside em [Pra] e Kohonen em [Koh-b].

• **Perceptron**

As redes Perceptron foram porpostas pelo psicólogo Frank Rosenblatt nos anos cinquenta. Elas são compostas, basicamente, por um neurônio com pesos sinápticos ajustáveis. O objetivo do Perceptron é classificar um conjunto de padrões em duas ou mais classes. Porém, para um desempenho adequado da rede, estes padrões devem ser linearmente separáveis.

No que diz respeito ao projeto das redes Perceptron, pode-se afirmar que eles são totalmente restritos ao problema a ser solucionado. Já que o número de neurônios de entrada e saída, que compõem a rede, determinará a capacidade de resolução do problema determinado.

As redes Perceptron multicamadas, da mesma forma que a simples camada e utilizando o como algoritmo de treinamento o Perceptron, são adequadas para solucionar problemas quando os dados são linearmente separáveis. Como este raramente é o caso, deve-se utilizar um método de pré-processamento para garantir que estes dados tornem-se linearmente separáveis antes de serem submetidos a entrada da rede neural.

• **Backpropagation**

O algorítmo *Backpropagation* foi criado a partir da generalização da regra de aprendizado de Widrow-Hoff [Wid-b]. As redes que usam este algorítmo se caracterizam por possuir múltiplas camadas e funções de transferência diferenciáveis não lineares.

Basicamente, nas redes *backpropagation*, cada entrada é multiplicada por um peso e estes produtos são somados para cada neurônio da rede. A esta somatória é aplicada uma função de ativação que produzirá o sinal de saída.

Os vetores de entrada e os correspondentes vetores de saída são utilizados no treinamento da rede, de modo elas possam aproximar uma função, associar vetores de entrada. De qualquer forma, a melhor adequação das rede *backpropagation* é para a generalização. Isto é, a partir de vários vetores de entrada, todos pertencentes a uma mesma

classe, a rede *backpropagation* é capaz de aprender as semelhanças mais significativa entre estes vetores, ignorando dados irrelevantes.

Sua estrutura consiste de uma camada de entrada, uma camada de neurônios escondidos e uma camada de saída. Quando três camadas não solucionam o problema, pode-se acrescentar camadas escondidas, o que implicará em um treinamento mais rápido da rede. O tamanho da camada de entrada é dependente da aplicação em questão, bem como o tamanho da camada de saída. Quanto ao número de neurônios na camada escondida, pode-se afirmar, somente, que ele deve consistir, inicialmente, de uma pequena fração do número de neurônios na camada de entrada. Se a rede não convergir, deve-se aumentar o número de nós, o mínimo possível para que a rede convirja.

- **Hopfield**

As redes de Hopfield possuem uma realimentação de suas saídas para suas entradas e, por isso, são denominadas redes recorrentes. Essa realimentação permite que a rede assuma diferentes comportamentos, na tentativa de chegar a resultados proveitosos. Essa dinâmica, porém, traz o inconveniente da instabilidade, podendo a rede passar de um estado a outro interminavelmente, sem produzir uma saída útil [Sep].

O objetivo de projeto de uma rede Hopfield é alcançar um conjunto de pontos de equilíbrio, tal que, quando uma condição inicial for fornecida a rede tenda a um destes pontos.

No caso da matriz de pesos sinápticos ser simétrica, pode-se determinar uma função de Lyapunov para o conjunto de equações diferenciais não lineares acopladas, que descrevem a rede. Esta função garante a convergência da rede para algum mínimo local, independente de seu estado inicial. Como a função de Lyapunov é formulada em termos da função objetivo a ser minimizada, o conjunto de equações direnciais resultante possuirá estados estáveis correspondentes ao mínimo local desta função.

• **Máquina de Boltzman**

Na máquina de Boltzmann, a saída dos neurônios individuais é uma função estocástica das entradas e não uma função determinista. Neste caso, a saída de um dado nó é calculada usando probabilidades, ao invés de um limiar ou função sigmóide.

A função da rede Boltzmann é aprender um conjunto de padrões de entrada e, então, estar apta a suprir perdas de partes dos padrões quando um padrão de entrada parcial ou ruidoso é processado.

A arquitetura da rede Boltzmann possui duas camadas de unidades: uma camada visível e uma intermediária. A rede é completamente interconectada entre camadas e entre unidades de cada camada. As conexões são bidirecionais e os pesos são simétricos.

Métodos estatísticos de treinamento provocam mudanças pseudo-aleatórias nos valores dos pesos, retendo aquelas mudanças que resultam em melhoramentos. Porém, o processo de aprendizagem da rede Boltzmann é muito lento. Pequenas redes podem requerer milhares de ciclos de processamento para aprender um conjunto de padrões de entrada adequadamente.

• **Mapas de Características**

O mapa de características auto-organizável foi proposto por Kohonen e aparenta uma rede neural laminada, cujas células são especificamente ajustadas a vários padrões de entrada ou classes de padrões, através de um processo de aprendizado não supervisionado. Por uma rede auto-organizável, entede-se que ela tenha a capacidade de usar experiências passadas para se adaptar a mudanças imprevisíveis em seu ambiente, sem auxílio externo.

Nestas redes, uma única célula ou um grupo de células, por vez, provê uma resposta ativa para a entrada corrente, como se cada um fosse um decodificador separado para a mesma entrada. Desta forma, um mapa ou sistema de coordenadas é ordenado e contém as respostas para as diferentes características de entrada. A ordenação nos mapas de características é obtida através de uma simples regra de atualização. Porém, o estudo do

processo auto-organizável tem se baseado em simulações por computador e não em resultados matemáticos.

O desempenho de uma mapa depende muito do número de passos de aprendizado durante o processo de ordenação. Assim, quanto maior o número de passos, melhor o desempenho do mapa. De forma empírica, Kohonen sugeriu que o número de passos de aprendizado deve ser em torno de 500 vezes o número de unidades da rede [Kar].

Os mapas de características auto-organizáveis têm sido utilizados para tarefas como robótica, reconhecimento de padrões e controle de processos [Kar].

3.8. SUMÁRIO

Este capítulo abordou as redes neurais artificiais a partir de um resumo de seu histórico. Procurou-se esboçar uma idéia geral a respeito do que são as RNAs e de como funcionam, citando-se as propriedades que justificam sua utilização. Além disso, foram apresentados alguns parâmetros que definem um modelo de RNA, como arquitetura, número de camadas, algoritmo de treinamento, entre outras. Por fim, foram apresentadas algumas redes neurais conhecidas, relevantes ao reconhecimento de fala.

O capítulo seguinte apresentará o desenvolvimento do sistema de reconhecimento de fala, descrevendo todos os procedimentos adotados em sua implementação. Além disso, será descrita a aplicação desenvolvida, que consiste na interação do sistema de reconhecimento de fala com um robô.

Capítulo 4

IMPLEMENTAÇÃO DO SISTEMA DE RECONHECIMENTO DE FALA

4.1. INTRODUÇÃO

Neste capítulo será apresentada a estrutura computacional proposta para o sistema de reconhecimento de fala, onde serão definidos todos os módulos que o compõe e descrita sua dinâmica de funcionamento, resultante da interação entre estes módulos. Por fim, será apresentada a célula flexível de manufatura utilizada como uma aplicação do sistema de reconhecimento de fala.

4.2. DESCRIÇÃO DO SISTEMA DE RECONHECIMENTO DE FALA IMPLEMENTADO

O sistema de reconhecimento de fala proposto tem sua estrutura ilustrada na Figura 4.1.

Este sistema foi idealizado de modo a reconhecer palavras isoladas e ser adaptativo ao locutor. O sistema foi implementado utilizando-se a Linguagem C [Bor], versão 4.5, sobre o sistema operacional Windows95 [Pet], programada como uma aplicação DOS.

A seguir, os módulos da Figura 4.1 serão detalhados, explicitando-se as funções implementadas neste trabalho.

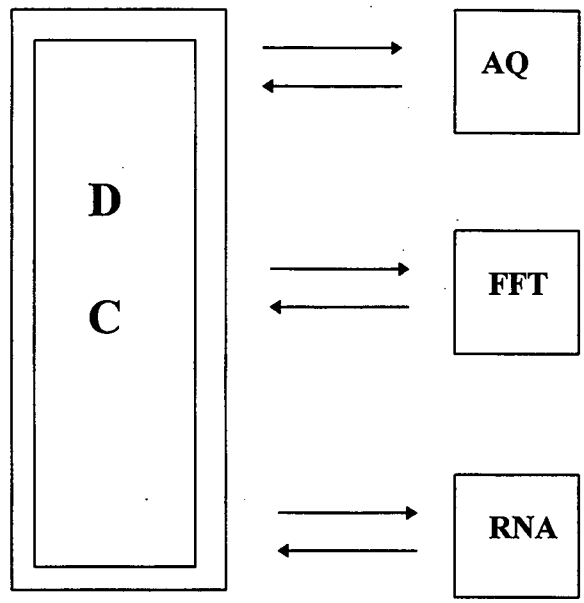

Figura 4.1. Estrutura computacional do sistema de reconhecimento de fala proposto

4.2.1. Módulo de Aquisição (AQ)

Para que dados em geral possam ser manipulados por uma máquina digital, é necessário que estes sejam quantizados e amostrados. A este procedimento, dá-se o nome de digitalização. O processo de quantização consiste em converter o sinal analógico para a forma digital através da subdivisão de uma faixa de entrada em classes de intervalos igualmente espaçados. Já o processo de amostragem consiste em medir funções contínuas em tempos discretos, levando-se em consideração o teorema de Nyquist a fim de obter-se uma representação coerente do sinal.

Para ser possível o processamento do sinal de áudio, foi utilizada uma placa *Sound Blaster* (SB16) [DKS]. Dentre as funções desempenhadas por esta, foram utilizadas as seguintes:

- Digitalizar o sinal de áudio
- Gravar e reproduzir o sinal digitalizado
- Controlar o volume do sinal

O diagrama de blocos que representa a placa de áudio é mostrado na Figura 4.2.

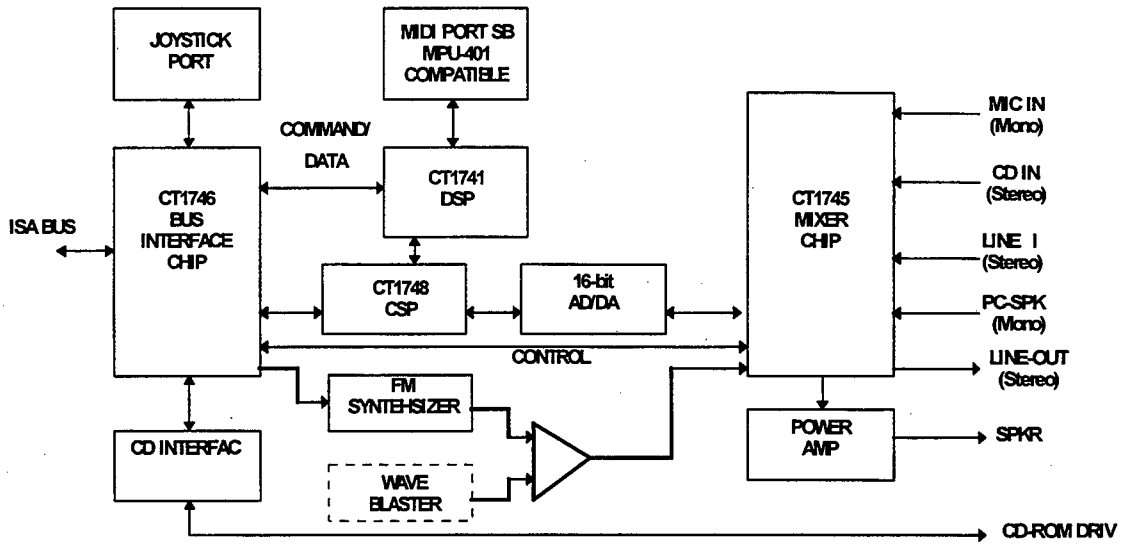

Figura 4.2. Diagrama de blocos da Sound Blaster 16

Um microfone com capacidade de filtragem para faixa de voz foi ligado a porta MIC IN do *chip* MIXER CT1745 (Figura 4.2). Através deste *chip* pôde-se controlar o volume do sinal de fala. A seguir, o sinal foi digitalizado. Para gravar e/ou reproduzir o sinal digitalizado foi utilizado o DSP[3] (*Digital Sound Processor*) CT 1741 (Figura 4.2).

4.2.1.1. Formatos de representação do sinal pelo DSP

Existem dois formatos de representação do sinal pelo DSP. Estes formatos são o 8-bit PCM e o 16-bit PCM. A Tabela 4.1 ilustra as características das duas representações.

Formato	Valor Máximo	Valor Mínimo	Valor do ponto médio
8-bit PCM	255 (0xFF)	0	128 (0x80)
16-bit PCM	32767 (0x7FFF)	-32768 (-0x8000)	0

Tabela 4.1. Formatos de representação do sinal pelo DSP

[3] Não se refere ao Processador Digital de Sinais e sim a um Processador Digital de Som

A representação utilizada foi a de 8-bit PCM. Esta representação foi escolhida por proporcionar menor utilização de memória que a representação 16-bit. Além disso, a 8-bit é suficiente para a quantização, o que foi detectado com base na literatura especializada, como em [Koh-b] [Lip-b].

4.2.1.2. Modos de transferência do sinal pelo DSP

O DSP está apto a trabalhar com variados modos de transferência das amostras do sinal. A Tabela 4.2 ilustra todos estes modos. Para escolher o modo de transferência toma-se como base um conjunto de características, entre elas: taxa de transferência, formatos de representação, número de canais e quantidade das amostras.

Modos de Transferência
8-bit Mono PCM Direct
8-bit Mono PCM Single-cycle
8-bit Mono PCM Auto-initialize
8-bit Mono ADPCM Single-cycle
8-bit Mono ADPCM Auto-initialize
8-bit/16-bit Mono PCM Single-cycle
8-bit/16-bit Mono PCM Auto-initialize
8-bit/16-bit Stereo PCM Single-cycle
8-bit/16-bit Stereo PCM Auto-initialize

Tabela 4.2. Modos de transferência suportados

O modo de transferência adotado neste trabalho foi o 8-bit PCM Direct. Neste modo os dados são representados no formato PCM 8 bits sem sinal. A tranferência dos dados se dá diretamente da entrada para o DSP e do DSP para a saída, não existindo um *buffer* intermediário. Será visto na seção 4.3 que as amostras são armazenadas no próprio *buffer* do decodificador. Como a transferência é feita diretamente, a taxa na qual ela ocorre é

determinada pela aplicação em questão, ao contrário dos outros modos, nos quais o usuário deve definir a taxa desejada previamente.

4.2.1.3. Funções implementadas sobre o DSP

As funções implementadas sobre o DSP são: ***read, write e reset.*** Essas funções fazem uso dos endereços de entrada e saída selecionados sobre a placa de áudio. A Tabela 4.3 ilustra todos os endereços das portas de entrada e saída do DSP [DKS].

Reset	2x6h	Usada para levar o DSP ao seu estado *default*
Read Data	2xAh	Usada para acessar dados dentro do limite do DSP
Write Data	2xCh	Usada para enviar comandos ou dados para o DSP
Write-Buffer Status	2xCh	Indica se o DSP está pronto para aceitar comandos ou dados
Read-Buffer Status	2xEh	Indica se há qualquer dado disponível para leitura

Tabela 4.3. Portas de entrada e saída do DSP

A função ***read*** é responsável por ler um dado do DSP. Para isto, os seguintes procedimentos são realizados:

1. Ler a porta *Read-Buffer Status*;
2. Se o bit 7 da porta lida for "1", então existe dado a ser lido, caso contrário não existe dado disponível na porta *Read Data*;
3. Ler o dado na porta *Read Data*, caso este exista.

A função ***write***, por sua vez, tem como objetivo escrever dados no DSP. Para isto, foram implementados os seguintes procedimentos:

1. Ler a porta *Write-Buffer Status*;
2. Se o bit 7 da porta lida for "0", então um dado pode ser escrito, caso contrário, a porta *Write Data ainda não está disponível;*

3. Escrever um dado na porta *Write Data,* caso este exista.

A função ***reset*** é responsável por inicializar o DSP e retorná-lo ao seu estado padrão de operação *(default).* Para que isso seja possível, os seguintes procedimentos são executados:

1. Escrever "1" na porta Reset e esperar por 3 μs;
2. Escrever "0" na porta Reset;
3. Observar se o byte 0AAh foi lido, a partir da porta Read Data. A porta Read-Buffer Status deve ser verificada para garantir que há dados antes de ler a porta Read Data.

Na programação do DSP, a função *reset* deve ser a primeira a ser chamada. Tipicamente, o processo de inicialização demora aproximadamente 100 μs [DKS]. Depois deste período de tempo, se não for detectado o valor 0AAh significa que houve algum erro de inicialização. Desta forma, esta função implementa um monitor de *timeout* para garantir que, se houver problemas de inicialização, a mesma será finalizada.

4.2.1.4. Comandos do DSP

A partir das funções descritas no item 4.2.1.3 é possível interagir com o DSP. Esta interação é realizada através da utilização de comandos do DSP pelas funções implementadas. Estes comandos são apresentados na Tabela 4.4.

Comando	Descrição
10h	Reproduz
D1h	Aciona o locutor
80h	Pausa na gravação

Tabela 4.4. Comandos do DSP

4.2.2. Transformada Rápida de Fourier (*Fast Fourier Transform*)

Para realizar a análise do sinal (item 2.3.2.3) foi utilizado o algoritmo da Transformada Rápida de Fourier (FFT). Este algoritmo foi baseado na Transformada de Fourier (FT) e será apresentado no decorrer desta seção.

Considerando-se um determinado sinal, pode-se fazer uma estimativa de seu espectro de potência, em um intervalo finito, utilizando-se a FT (Equação 4.1).

$$F(e^{jwT}) = \sum_{n=-\infty}^{+\infty} f(nT)e^{-jwnT} \qquad \text{(Equação 4.1)}$$

A Equação 4.1 fornece o espectro de Fourier resultante da amostragem de um sinal contínuo f(t), gerando um número infinito de valores da variável discreta f(nT). Do ponto de vista prático, necessita-se que esse espectro possua um número finito de pontos. Assim, deve-se truncar a somatória da Equação 4.1 e expressá-la na forma ilustrada pela Equação 4.2:

$$\hat{F}(e^{jwT}) = \sum_{n=0}^{N-1} f(nT)e^{-jwnT} \qquad \text{(Equação 4.2)}$$

A partir deste truncamento, obtém-se como resultado uma estimativa do espectro de Fourier, $\hat{F}(e^{jwT})$. Quanto maior for o número de pontos especificados por N, mais $\hat{F}(e^{jwT})$ se aproxima de $F(e^{jwT})$.

O espectro total de Fourier consiste de uma faixa contínua de freqüências w, repetidas periodicamente. Na prática, pode-se estipular um número finito de valores de w para os quais pode-se calcular $\hat{F}(e^{jwt})$. Para isso, sendo a freqüência de amostragem w_s dada por $\frac{2\pi}{T}$, corta-se a faixa de freqüências entre 0 e w_s em N valores, de modo a obter-se o mesmo número de amostras no tempo. Desta forma, estes valores ficam espaçados na freqüência de $\frac{2\pi}{NT}$.

Portanto, o espectro de Fourier é analisado a cada K($\frac{2\pi}{NT}$) valores de freqüência onde K é o índice associado com cada freqüência discreta escolhida. Então, a Equação 4.3 é expressa como:

$$\hat{F}(e^{jk(2\pi/NT)T}) = \sum_{n=0}^{N-1} f(nT)e^{-jk(2\pi/NT)nT} \qquad \text{(Equação 4.3)}$$

Já que o número de freqüências analisadas é determinado pelo número de amostras no comprimento do sinal escolhido, então é mais usual escrever a expressão (Equação 4.3) em sua forma compacta, chamada de Transformada Discreta de Fourier (Equação 4.4).

$$F(k) = \sum_{n=0}^{N-1} f(n)e^{-j[(2\pi nk)/N]} \qquad \text{(Equação 4.4)}$$

Na Equação 4.4, para o cálculo do espectro de Fourier, a multiplicação do valor da amostra f(n) pelo termo exponencial é feita para N amostras e N freqüências, portanto, existem N^2 multiplicações envolvidas neste processo. Este procedimento foi revisto por Cooley [Coo], que propôs um algoritmo que otimiza a implementação da Equação 4.4, de forma que o número de multiplicações é reduzido para N. $\log_2$N. Este algoritmo é conhecido como Transformada Rápida de Fourier (*Fast Fourier Transformer*).

Com os resultados obtidos da FFT, pode-se chegar ao espectro de potência calculando-se o módulo dos coeficientes resultantes.

4.2.2.1. Janelamento

A truncagem da Transformada Discreta de Fourier em um número finito de pontos é equivalente a observar o sinal original f(t) através de uma "janela" de tempo w(t), como ilustrado na Figura 4.3. Matematicamente, este processo é expresso pela Equação 4.5.

$$f_w(t) = f(t).\ w(t) \qquad \text{(Equação 4.5)}$$

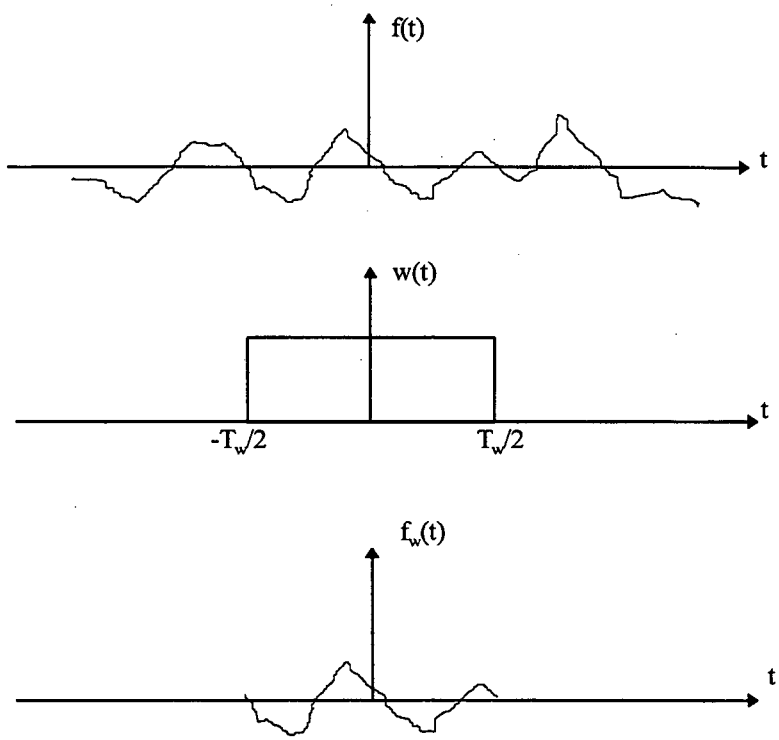

Figura 4.3. Multiplicação do sinal original f(t) pela função janela retangular w(t)

É importante notar que, no domínio da freqüência, o efeito do janelamento altera o conteúdo espectral do sinal a ser analisado.

Formalmente, considerando-se a transformada de Fourier do sinal F(w) e a transformada de Fourier da janela W(w). Pode-se escrever a Equação. 4.5 na forma da Equação 4.6.

$$f_w(t) = \left[\frac{1}{2\pi}\int_{-\infty}^{+\infty} F(w_1)e^{jw_1 t}dw_1\right]\left[\frac{1}{2\pi}\int_{-\infty}^{+\infty} W(w_2)e^{jw_2 t}dw_2\right] \qquad \text{(Equação 4.6)}$$

A transformada de Fourier de $f_w(t)$ pode, portanto, ser escrita como na Equação 4.7.

$$F_w(w) = \frac{1}{(2\pi)^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(w_1)W(w_2)e^{j(w_1+w_2)t}e^{-jwt}dw_1 dw_2 dt \qquad \text{(Equação 4.7)}$$

Esta relação sugere, formalmente, o modo no qual o espectro do dado original F(w) se modifica para uma "janela espectral" W(w). Assim, conseqüentemente, as componentes

do espectro de energia do sinal ficam distorcidas. Este efeito é como se o espectro ficasse "borrado".

Sabe-se que quanto maior for o comprimento da janela (T_w) em relação à freqüência do sinal analisado, menor é a interferência de uma componente de freqüência sobre a outra. Assim, tem-se estudado tipos de "janelas" para serem inseridas propositadamente no processo de análise, permitindo a visualização do sinal de maneira que a influência destas seja positiva. Ou seja, o objetivo final do processo de "janelamento" é concentrar a energia do sinal.

Dentre as principais "janelas" temos:

- Hann ou Hanning
- Blackman
- Hamming
- Kaiser

A "janela" utilizada neste trabalho é a de Hamming, pois este tipo de "janela" concentra 99.26% da energia espectral no lóbulo principal de sua transformada e o restante nos lóbulos laterais. Desta forma, a influência de uma freqüência na outra deverá ser mínima.

4.2.3. Rede Neural Artificial (RNA)

As redes neurais artificiais foram adotadas para o reconhecimento de fala, neste trabalho, por terem a capacidade de classificar.

Desta forma, no sistema de reconhecimento de fala implementado, a rede neural artificial atua como o modelo acústico (item 2.3.2.5).

O modelo projetado para o sistema foi definido por uma rede *Perceptron* multicamadas (MLP) com algoritmo de treinamento *Backpropagation*. A camada de entrada contém dezesseis neurônios, a camada de saída, quatro e a camada intermediária, sessenta neurônios (Figura 4.4).

Uma rede *Feedforward* foi escolhida devido a sua estrutura apresentar maior tendência à estabilidade do que as redes *Feedback*, que podem ser instáveis devido a realimentação. A instabilidade do algorítmo de treinamento nas redes *Feedforward* poderá surgir, mas será devida a outros parâmetros que não a sua estrutura.

Outra questão que influenciou a escolha da estrutura *Feedforward* é que, apesar de não haver na literatura especializada indícios da maior adequação desta sobre a *Feedback*, não há nada que a contra-indique para aplicações de reconhecimento de fala.

Em relação a escolha do algoritmo de treinamento da rede, foi tomado como base o fato de que o algoritmo *Backpropagation* é de propósito geral. Assim, de certa forma, pode-se esperar que obtenha-se resultados satisfatórios no aprendizado, mesmo sabendo-se dos problemas de tempo de convergência e mínimo local que este algoritmo pode apresentar.

Além disso, o *Backpropagation* é utilizado em alguns trabalhos sobre reconhecimento de fala, presentes na literatura especializada. Entre os autores, podemos citar Lippman em [Lip-c].

A escolha de uma composição multi-camadas foi baseada no fato de que uma rede multi-camadas com função de transferência não linear diferenciável, pode estabelecer uma relação - linear ou não linear - entre a entrada e a saída da rede. A utilização de uma rede simples camada implicaria em uma correspondência limitada entre a entrada e a saída.

Assim, utilizou-se uma estrutura com três camadas composta de neurônios com funções de transferência não lineares (sigmóide) e, somente na camada de saída, neurônios com funções de transferência lineares [Rum-a]. Isto porque a função sigmóide possui uma faixa de operação que varia entre zero e um, limitando a faixa de valores que podem ser obtidos na saída da rede. Com a utilização de neurônios com funções de transferência lineares na última camada a rede pode trabalhar com qualquer valor em sua saída.

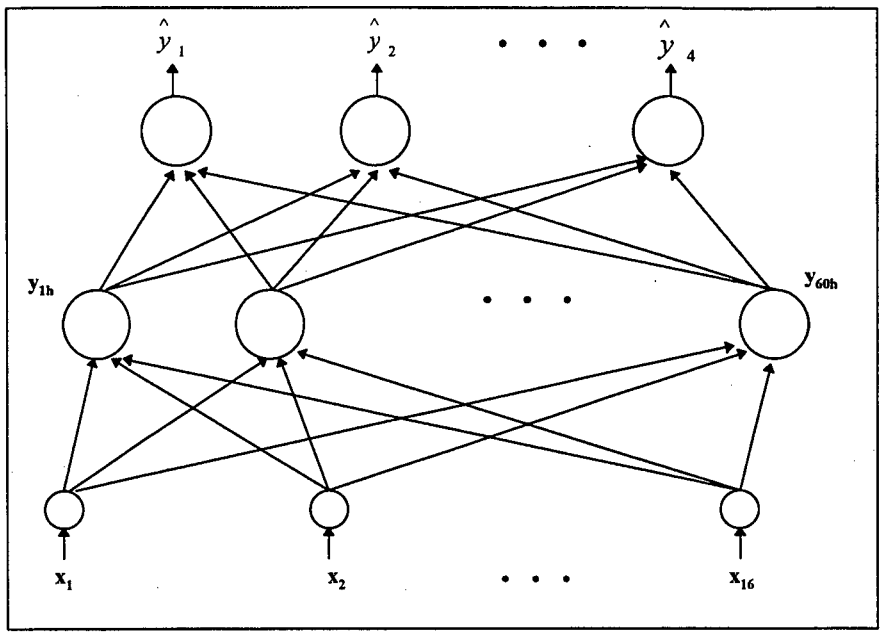

Figura 4.4. Estrutura da rede neural implementada

Esta estrutura final foi obtida depois de alguns testes, já que não existe, até o presente momento, um procedimento padrão para o projeto de redes neurais artificiais. Os testes realizados serão comentados no Capítulo 5.

Em relação às questões de implementação da rede, foi utilizada a programação seqüêncial por questões de simplicidade.

Como esperava-se que, no decorrer da implementação, houvesse inúmeras mudanças na arquitetura da rede, procurou-se projetá-la da maneira mais modular possível, para que todas as modificações necessárias fossem possíveis. Assim, todas as questões envolvidas na configuração da rede ficam em um arquivo à parte àquele contendo suas rotinas e estruturas de dados.

Além disso, a adoção de múltiplos arquivos de forma modular possibilita que o algoritmo de treinamento possa ser substituído com facilidade, caso isso seja necessário.

A estrutura de dados adotada na implementação da rede foram as matrizes. Não foram utilizadas as listas encadeadas porque, apesar de possibilitarem a alocação dinâmica de memória, elas apresentam duas desvantagens em relação às primeiras: demandam um consumo excessivo de memória devido ao número de apontadores necessários para encadear

a estrutura de uma rede neural e apresentam uma taxa de processamento muito baixa para propagação do sinal [Fre].

Os pesos e limitadores são dados que caracterizam o **estado da rede**. Estes dados são todos armazenados em arquivos de forma que, a partir destes, o estado da rede possa ser recuperado sem a necessidade de um novo treinamento. Após o início do programa estes valores são buscados nos arquivos e novamente armazenados em suas respectivas estruturas de dados.

Duas matrizes são responsáveis por armazenar os dados da rede: a matriz de pesos *Weigh* e a matriz de limitadores *Threshold.*

Duas funções foram implementadas para acessar o dados que caracterizam o estado da rede: *load* para carregar os dados a cada início de execução de um teste e *save* para salvá-los após o treinamento.

Os dados de entrada e saída são também armazenados em arquivos, mas não possuem qualquer estrutura de dados para armazená-los. Eles são buscados destes arquivos no decorrer da execução do programa.

Neste módulo, as principais funções implementadas foram

Train: é utilizada para treinar a rede, fazendo o cálculo direto e a retro-propagação do erro para a entrada da mesma.

Test: é utilizada para testar a rede, simulando-a através do cálculo direto.

Error: é utilizada pela função *Train* para o cálculo do erro entre a saída atual, proporcionada pela rede, e a saída desejada.

Assim, o módulo RNA é utilizado para realizar o treinamento, bem como testar o resultado do mesmo. O teste pode ser feito imediatamente à etapa de treino ou posteriormente, em outra execução.

4.2.4. Módulo Decodificador (*DC*)

Essencialmente, o Decodificador é responsável pelo gerenciamento das tarefas executadas pelo sistema. Assim, a partir dele são feitas várias solicitações de execução de funções de outros módulos, como será descrito na seção 4.3.

Dentre as funções do Decodificador, podemos citar: construção de um *buffer* com as amostras do sinal de fala, segmentação destas amostras, montagem dos quadros de fala, comparação dos quadros de fala com o modelo acústico, entre outras.

4.3. DINÂMICA DO SISTEMA DE RECONHECIMENTO DE FALA

Nesta seção, será descrita toda a dinâmica do sistema de reconhecimento de fala, que resulta da interação entre os módulos anteriormente citados.

A interação do módulo DC com os módulos AQ, FFT e RNA é feito através de chamadas as suas funções. Os procedimentos de análise do sinal (quadro de fala), análise acústica (quadro de pontos) e gramática são realizados no decorrer da execução do sistema.

A princípio é feita a aquisição do sinal. O DC deve interagir com o módulo AQ, solicitando que o mesmo inicialize o DSP através de uma chamada a função *Reset*. Em seguida, o módulo AQ deverá monitorar a entrada do *chip* MIXER, o qual estará conectado a um microfone. Quando houver um sinal de fala na placa de aquisição, o DSP deverá digitalizá-lo. Após detectado o início de uma palavra, o módulo DC solicita ao módulo AQ que inicie a gravação do sinal digitalizado, armazenado-o num *buffer*. A função *read* será responsável pela leitura dos dados do DSP. Quando detectado o final de uma palavra encerra-se o processo de gravação.

A seguir, tem início o processo de análise do sinal. De posse dos dados no *buffer*, o DC segmenta estes dados em N partes, sendo que N indica o número de pontos desejados no cálculo do FFT. Neste caso, o valor de N adotado foi 256. Em seguida, o módulo DC solicita ao módulo FFT o "janelamento" e o cálculo da transformada para cada segmento da palavra, o que é feito através das funções *Window* e *FFT*, respectivamente. Após este processo, o DC calcula, a cada 16 coeficientes, a média dos mesmos para montar os quadros de fala.

O processo de análise acústica tem início com a normalização dos quadros de fala que, posteriormente, são confrontados com o modelo acústico gerado pelo módulo RNA. Isto é realizado pelo DC que, com os resultados desta comparação constrói o quadro de pontos.

A última etapa a ser descrita é a de reconhecimento. De posse do quadro de pontos e do autômato que representa a gramática, identifica-se qual o comando decodificado a partir da palavra pronunciada.

Uma descrição resumida da dinâmica do sistema é ilustrada na Figura 4.5.

1°. Aquisição do sinal:
- **Reset DSP**
- **Captura uma palavra, detectando início e fim**
- **Armazena as amostras em um *buffer***

2°. Análise do sinal:
- **Tira-se o FFT da palavra segmentada**
- **Monta o quadro de fala**

3°. Análise acústica:
- **Compara o quadro de fala com o modelo acústico**
- **Monta o quadro de pontos**

4°. Reconhecimento:
- **Baseado no quadro e na gramática decide qual é a palavra**

Figura 4.5. Descrição resumida da dinâmica do sistema.

4.4. SOLUÇÕES ADOTADAS NA IMPLEMENTAÇÃO DO SISTEMA

Nesta seção, serão abordados alguns problemas encontrados na implementação do sistema de reconhecimento de fala e as soluções adotadas para os mesmos.

4.4.1. Aquisição sem a utilização de um *buffer* intermediário

Para fazer a aquisição do sinal de fala, vários testes foram feitos com a placa de áudio.

São diversos os modos de transferência de dados oferecidos pela placa de áudio. A maioria destes modos utiliza uma memória intermediária, que resulta em maior velocidade de aquisição. Porém, no caso deste trabalho, o *buffer* intermediário não seria adequado. Isto porque sua capacidade de 8 Kbytes não seria suficiente para suportar uma palavra completa. Assim, um segundo *buffer* seria necessário para armazenar os fragmentos de uma determinada palavra, conforme eles chegassem ao *buffer* intermediário, até que a palavra estivesse completa e pronta para ser analisada. Neste caso, o *buffer* intermediário atua negativamente, consumindo memória extra e retardando o processo de reconhecimento.

A solução adotada para este problema consiste na não utilização do *buffer* intermediário. Para isso, a placa foi reprogramada para utilizar o modo de transferência direto. Além disso, um *buffer* próprio foi implementado, de tamanho maior e suficiente para receber uma palavra com, no máximo, 3 s de duração, com freqüência de amostragem de 16 KHz (32 Kbytes).

4.4.2. Detecção do início e do fim de uma palavra

No reconhecimento de palavras, utiliza-se um *buffer* de tamanho limitado para armazenar a palavra a ser analisada. Porém, este *buffer* sempre é sub ou super-estimado, já que não se sabe, *a priori*, o tamanho exato da palavra. Além disso, dependendo da velocidade com que a palavra é pronunciada, pode-se armazená-la inteira ou parcialmente no *buffer*, necessitando-se, desta forma, de um procedimento para detecção do início e do fim de uma palavra.

Existem vários métodos para detecção de início e fim de uma palavra. O método implementado neste trabalho foi baseado na amplitude do sinal durante uma determinada faixa de tempo. Assim, para identificar os parâmetros de amplitude e duração, foram feitas várias observações do sinal de fala até que se chegasse a um valor estimado destes parâmetros. Assim, os sinais cuja amplitude e duração estivessem abaixo do valor estimado eram considerados ruído e, desta forma, eram utilizados como delimitador de início e fim das palavras.

4.4.3. Análise do sinal segmentado

Além do *buffer* necessário para aquisição (item 4.4.1), para fazer a análise do sinal total de uma palavra com 3 s de duração seria necessário um outro *buffer* de 32 Kbytes. No caso deste trabalho, esse consumo de memória seria demasiado, não restanto o suficiente para o processamento da rede neural.

Assim, adotou-se como solução dividir o sinal em 256 segmentos, de forma que o *buffer* pudesse ter suas dimensões também reduzidas. A análise do sinal de fala é realizada sobre cada segmento, levando-se em consideração a característica do sinal de possuir um comportamento estacionário a cada 10 ms.

4.5. DESCRIÇÃO DA APLICAÇÃO: O ROBÔ

Implementado o sistema de reconhecimento de fala, utilizou-se um robô como aplicação. O objetivo é verificar o funcionamento adequado do sistema implementado bem como demonstrar sua utilidade.

O robô utilizado encontra-se no Laboratório de Controle e Microinformática (LCMI - LAI) e possibilita o desenvolvimento de trabalhos e pesquisas nas áreas de automação e manufatura. Este robô faz parte de uma célula flexível de manufatura, cuja estrutura emula, de forma miniaturizada, as células de manufatura encontradas atualmente na indústria.

Fabricada pela Allen-Bradley Co. [All], a célula é composta, dentre os equipamentos utilizados, por aqueles apresentados na Tabela 4.5.

Quantidade	Descrição
15	Fotocélulas
14	Sensores Fim de Curso
29	Válvulas pneumáticas de acionamento elétrico associadas aos respectivos pistões
01	Robô pneumático Pick & Place
04	Cintas transportadoras acionadas por motores DC
01	Compressor de ar
09	Cilindros de alumínio
08	Pallets metálicos para transporte de cilindros
01	CLP modular composto de uma fonte, um processador e seis módulos de I/O montados sobre bastidores de 10 slots

Tabela 4.5. Equipamentos que compõem a célula flexível de manufatura

Estes equipamentos estão dispostos segundo o *layout* ilustrado na Figura 4.6.

4.5.1. Interação entre o sistema de reconhecimento de fala e a célula de manufatura

A implementação da aplicação consiste em estabelecer a comunicação entre o usuário e a célula de manufatura, através de comandos de voz. Assim, quando um locutor pronunciar ao microfone uma das palavras para as quais o sistema de reconhecimento foi treinado para identificar, o procedimento equivalente a este comando deve ser iniciado pela célula.

Em uma versão inicial, os comandos identificados pelo sistema e os respectivos procedimentos executados pela célula são ilustrados na Tabela 4.6.

Comando	Procedimento executado pela célula
"liga"	Acionar as cintas transportadoras
"desliga"	Desligar as cintas transportadoras
"abre"	Abrir a garra do robô
"fecha"	Fechar a garra do robô

Tabela 4.6. Comandos identificados pelo sistema de reconhecimento de fala

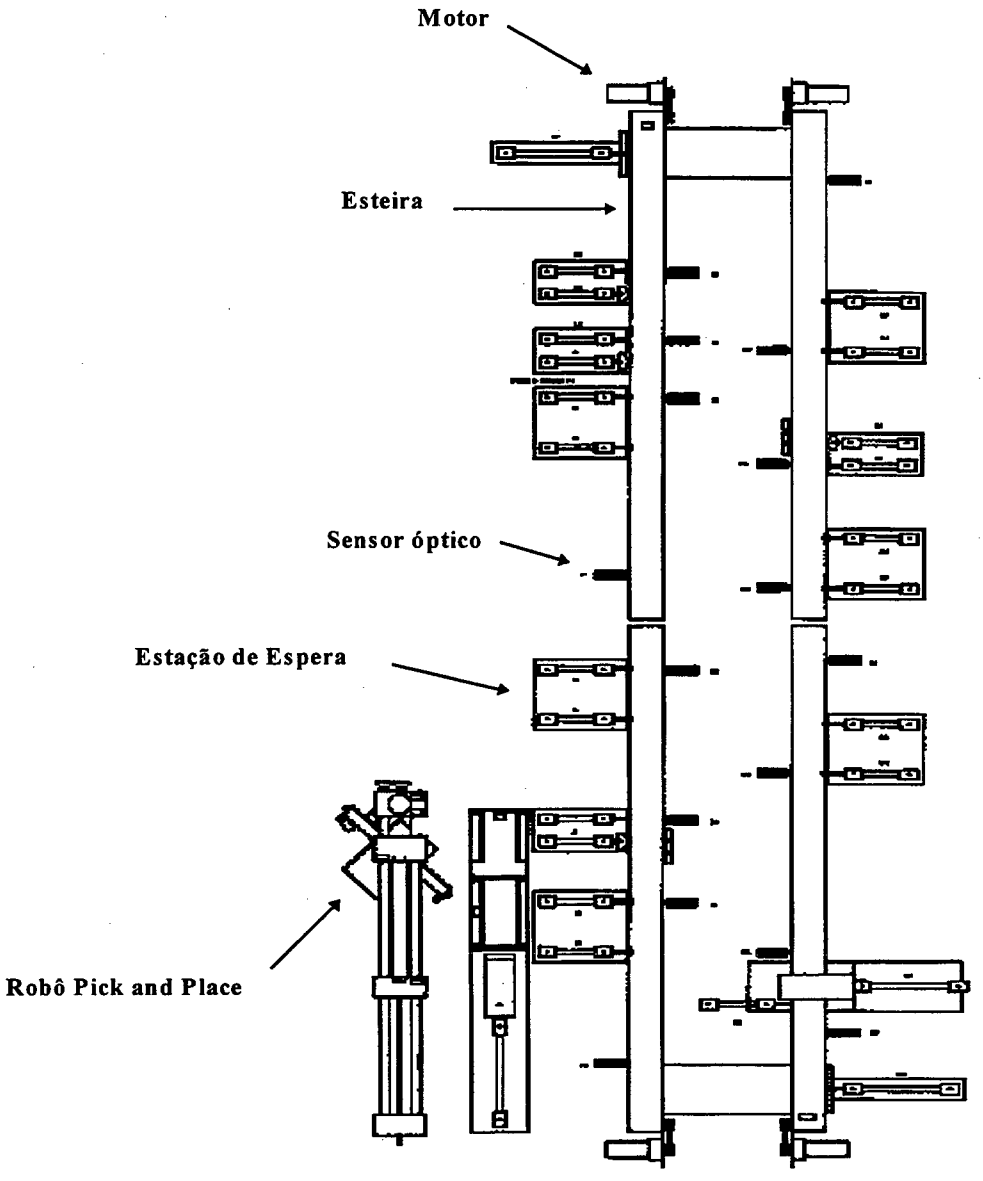

Figura 4.6. *Layout* da célula flexível de manufatura

A interação do sistema de reconhecimento com a célula de manufatura é realizada através da comunicação entre o computador (que executa o sistema de reconhecimento) e o

CLP (que controla a célula de manufatura). Esta comunicação se dá através do envio de um valor numérico, correspondente ao comando falado, para o CLP. Este por sua vez, interpreta o valor numérico recebido e aciona as funções correspodentes da célula flexível de manufatura.

Para que isto fosse possível, o CLP teve que ser previamente programado, o que foi realizado por Leal, em [Lea].

4.5.2. Comunicação do PC com o CLP

Para realizar a comunicação entre o microcomputador e o CLP, foi utilizada uma placa desenvolvida no LCMI-LAI, denominada CONCEL [Zil] [Lea]. A estrutura resultante desta integração é ilustrada na Figura 4.7.

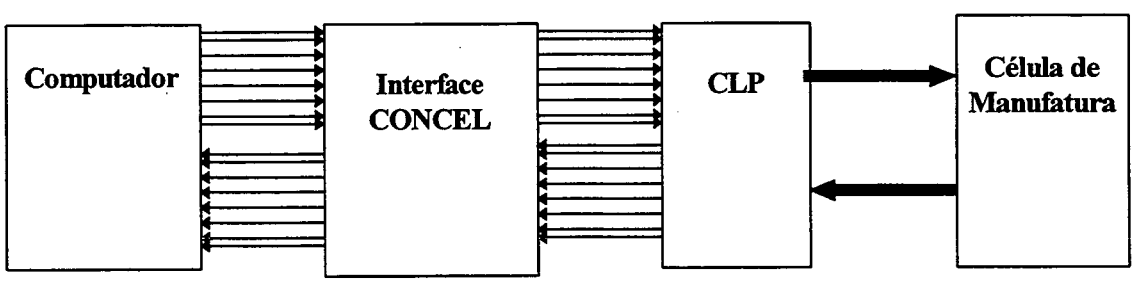

Figura 4.7. Integração entre o computador e a célula de manufatura

A interface é colocada entre o microcomputador, a partir de sua porta paralela e o CLP. Seu objetivo é tornar compatíveis os níveis de tensão dos sinais elétricos utilizados pelo computador, entre 0 e 5 V, com os sinais elétricos do CLP, entre 0 e 24 V.

Desta forma, o PC mantém em suas saídas os dados correspondentes ao último evento ocorrido, até que um novo evento seja enviado. Assim, torna-se necessário identificar se a informação existente na saída de dados é inédita ou se já foi processada. Isto é feito através do bit mais significativo do byte contendo o número do evento. Com os sete bits restantes será armazenado o número associado ao evento em questão. Desta forma, é possível enviar do PC para a célula 128 eventos (2^7).

A interface CONCEL fornece a via de acesso ao CLP. Porém, ainda é necessário um procedimento que, seguindo o protocolo estabelecido pelo CLP, possa enviar um valor numérico correspodente a uma determinada tarefa. Este procedimento é implementado neste trabalho utilizando-se mecanismos de chamadas a interrupções e é responsável por enviar um valor numérico à porta paralela, a qual está conectada à interface CONCEL.

4.6. SUMÁRIO

Neste capítulo foram abordadas as questões referentes ao sistema de reconhecimento de fala implementado, propriamente dito. Foram analisados os módulos que compõem o sistema computacional, bem como a interação entre eles. Além disso, foi detalhada a dinâmica resultante da integração dos módulos do sistema e apresentada a célula de manufatura, que foi utilizada como aplicação. Todos os procedimentos implementados para integrar a célula ao sistema de reconhecimento foram apresentados.

O próximo capítulo irá apresentar alguns dos resultados obtidos neste trabalho. Entre eles, os resultados referentes ao modelo adotado e a adequação do sistema à aplicação utilizada.

Capítulo 5

RESULTADOS DO SISTEMA DE RECONHECIMENTO DE FALA

5.1. INTRODUÇÃO

Neste capítulo serão relatados todos os principais resultados obtidos provenientes do desenvolvimento e implementação do sistema de reconhecimento de comandos verbais utilizando as técnicas de redes neurais artificiais. Os resultados referentes à aplicação deste sistema à automação também serão apresentados, pois a proposta deste trabalho envolve a aplicação do sistema desenvolvido a uma célula flexível de manufatura.

5.2. RESULTADOS A RESPEITO DO MODELO DE RNA ADOTADO

Existe uma gama de parâmetros que podem ser levados em consideração a fim de ter sua influência estudada no processo de reconhecimento. Desta forma, serão analisados os efeitos da variação de alguns destes parâmetros, tais como: número de entradas, normalização dos sinais de entrada fornecidos a rede, número de camadas, número de neurônios em cada camada, funções de transferência, etc. Por fim, serão avaliados os resultados do sistema como um todo, ou seja, o reconhecedor de comandos verbais aliado a célula flexível de manufatura. Os parâmetros relevantes, neste caso, dizem respeito a velocidade de resposta da célula de manufatura e eficiência. É importante lembrar que todos os experimentos utilizaram a rede *Perceptron* multi-camadas com algoritmo de treinamento *Backpropagation.*

5.2.1. Influência do número de entradas

Com relação ao número de entradas ou número de neurônios na camada de entrada da rede, foram feitos dois tipos de testes. O primeiro foi com uma rede composta de dezesseis neurônios na entrada, trinta e dois na camada intermediária e quatro na camada de saída. Para esta rede foram apresentados dois padrões de entrada, cada um com uma amostra, e obteve-se uma taxa de erro de 15 %.

No segundo experimento, foram adicionados mais neurônios à entrada, perfazendo um total de vinte e seis neurônios nesta camada. Com tal modificação, a taxa de erro atingida foi de 30 %.

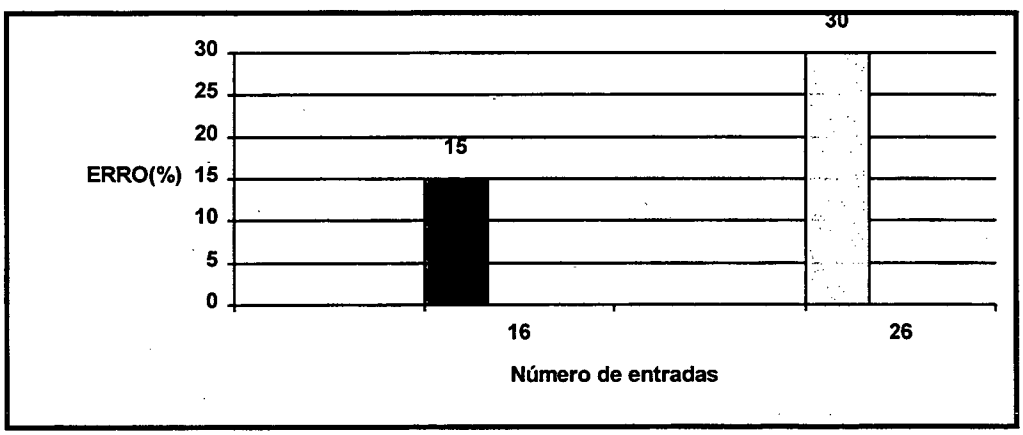

Figura 5.1. Resultados da variação do número de neurônios na camada de entrada

Desta forma, obteve-se um aumento da taxa de erro (Figura 5.1). Isto, supõe-se, é devido a não levar-se em consideração, na faixa utilizada, a janela de 10 ms, o que resulta em um comportamento variável maior do espectro. Além disso, é importante lembrar que nem sempre pode-se trabalhar com um número elevado de neurônios na camada de entrada, devendo-se levar em consideração a quantidade de memória e o tempo de processamento despendido pelo sistema que deseja-se implementar.

5.2.2. Influência da normalização dos dados de entrada

A faixa de valores de entrada pode não afetar o desempenho de uma rede, já que a rede pode aprender a compensá-los. Isto pode ser feito através de um deslocamento nos limitadores dos neurônios pertencentes às camadas intermediárias. Na prática, porém, sabe-se que os valores de entrada devem ser igualmente normalizados, de forma que, assim, a rede possa dar a mesma atenção a todas as entradas [Fre].

Assim, foram feitos testes para normalização nas faixas entre [0,1] e [-1,1]. Estes testes são expressos na Figura 5.2. Para isto foi utilizada uma rede com dezesseis neurônios de entrada, quatro de saída e dois na camada intermediária. Nos neurônios da camada de saída foram utilizadas funções de transferência lineares e nos neurônios das camadas restantes as funções de transferência sigmóides.

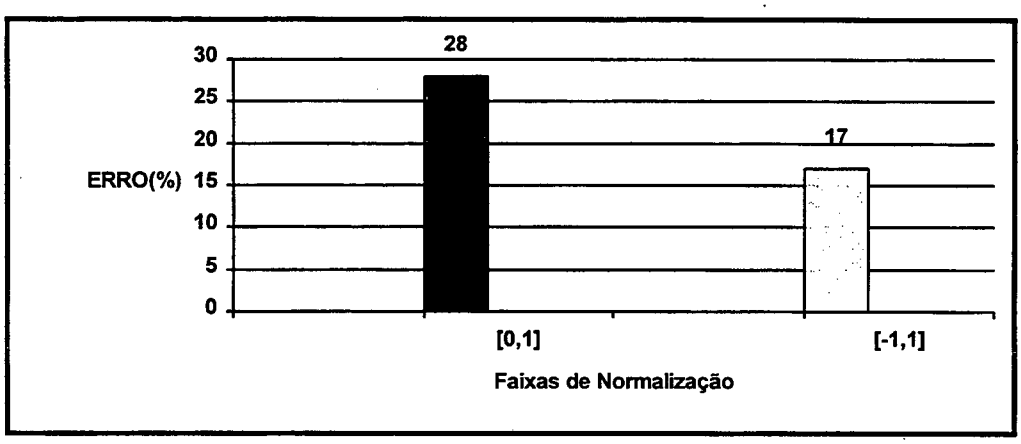

Figura 5.2. Normalização nas faixas entre [0,1] e [-1,1]

5.2.3. Influência do número de camadas

O primeiro parâmetro que surge quando se fala em projetar uma rede neural para servir como modelo acústico, são as suas dimensões. Atualmente, não se conhece uma metodologia analítica a ser seguida, de forma que se possa chegar a uma conclusão sobre estes fatores. Assim, utiliza-se a heurística como principal ferramenta para se fazer esta análise. Por isso, o dimensionamento da rede dependerá do problema a ser resolvido.

No projeto de uma rede neural, a primeira pergunta a ser feita é se a rede necessita ter camadas intermediárias ou não. Teoricamente, uma rede sem camadas intermediárias (*Perceptron* simples camada) pode formar somente regiões de decisão lineares, sendo garantido, porém, 100% de precisão no processo de classificação se o conjunto de treinamento é linearmente separável. De forma contrária, uma rede com uma ou mais camadas intermediárias (*Perceptron* multi-camadas), pode formar regiões de decisão não lineares.

A fim de avaliar-se o grau de importância da presença de uma camada intermediária, fez-se os seguintes testes: implementou-se uma rede com dezesseis neurônios na camada de entrada, quatro neurônios na camada de saída e nenhum neurônio na camada intermediária, caracterizando-se, assim, uma rede simples camada; então, apresentou-se a ela um padrão para treinamento e três padrões de teste, obtendo como taxa de erro para os padrões de treinamento 83 %.

A mesma rede, com uma alteração no número de neurônios na camada intermediária (um neurônio), caracterizando, assim, uma rede multi-camadas o mais simples possível, foi implementada a fim de ser testada e seus resultados confrontados com a primeira. Obteve-se uma taxa de 17 % de erro, ou seja, uma melhora significativa de uma estrutura para outra.

Este resultado confirma que uma rede multi-camadas está mais propensa a aprender quando comparada a uma rede simples camada. A Figura 5.3 ilustra os resultados obtidos do teste anteriormente citado.

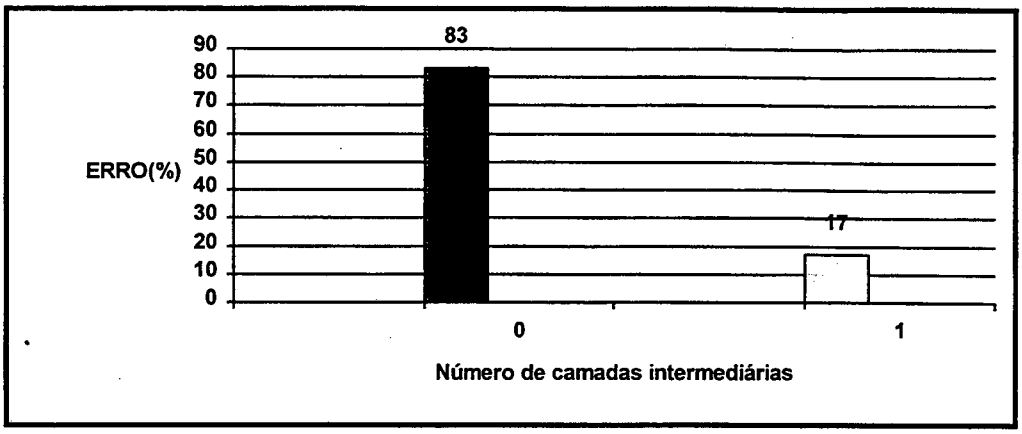

Figura 5.3. Resultados de uma rede simples camada *versus* multi-camadas

5.2.4. Influência do número de neurônios na camada intermediária

Outro fator analisado foi a influência do número de neurônios nas camadas intermediárias. Para isso, os seguintes testes foram realizados: usando uma rede com dezesseis neurônios de entrada e quatro de saída, variou-se o número de neurônios na camada intermediária e obtivemos como resultado os seguintes valores (Figura 5.4).

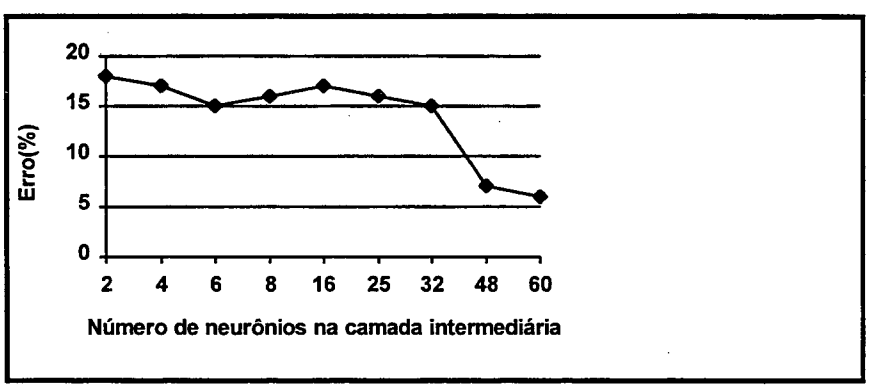

Figura 5.4. Influência do número de neurônios na camada intermediária

O número de neurônios nas camadas intermediárias tem um forte impacto no desempenho de uma rede MLP. Quanto mais unidades intermediárias a rede tem, mais superfícies de decisão complexas podem ser formadas. Portanto, esta rede poderá classificar com maior precisão, confirmando, assim, o resultado expresso na Figura 5.4.

5.2.5. Influência da Função de Transferência dos neurônios

A função de transferência ou função de "ativação" é um outro fator muito importante na escolha do modelo de neurônio a ser adotado. O único fator teórico disponível, que contribuiu para a escolha da função, foi a faixa na qual esta poderia operar. Assim, foram realizados testes para verificar qual função proporcionava um melhor resultado.

As funções que foram utilizadas no teste foram a sigmóide e sigmóide simétrica (Figura 3.3). A Figura 5.5 exprime os resultados alcançados para estas funções.

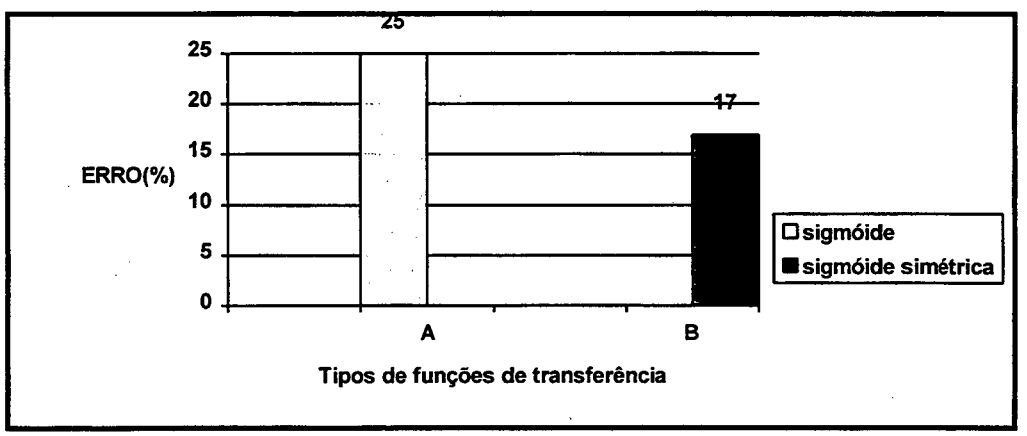

Figura 5.5. Função simóide *versus* função sigmóide simétrica

5.2.6. Influência da taxa de aprendizado

A taxa de aprendizado foi analisada visando diminuir o tempo despendido com treinamento. Isto porque quanto maior a taxa de aprendizado, menos tempo é consumido para que a rede neural artificial aprenda.

Assim, observou-se que não é tão simples dimensionar a taxa de aprendizado, pois esta é intimamente relacionada a outros fatores, alguns deles citados abaixo. Além disso, sabe-se que duas redes configuradas de maneiras iguais, inclusive com a mesma taxa de treinamento, podem obter resultados finais diferentes para um mesmo conjunto de padrões de teste.

Portanto, utilizou-se uma rede com dezesseis neurônios na camada de entrada, quatro na camada de saída e variou-se os da camada intermediária de 2, 4, 8, 16, 25, 32, 48 a 60 neurônios. A esta rede foram submetidas duas palavras, com dois padrões representando cada palavra. Em etapas diferentes vários valores de coeficientes de aprendizado foram utilizados, entre eles: 0.75, 0.55, 0.25, 0.075, 0.055, 0.025, 0.0075, 0.0055, 0.0025, 0.00075, 0.00055 e 0.00025.

Pôde-se observar que a taxa de aprendizado é influenciada principalmente pelos seguintes fatores: número de padrões de treinamento, normalização da entrada, função de transferência na saída (linear ou sigmóide) e número de neurônios.

Em relação ao número de padrões de treinamento, pôde-se observar que quanto maior o conjunto de treinamento, menor deve ser a taxa de aprendizado utilizada para que o algoritmo convirja mais rapidamente.

Em relação à normalização da entrada observou-se que, aumentando o desvio do padrão de entrada deve-se aumentar a taxa de aprendizado para compensar o fato de que as unidades intermediárias tornam-se saturadas. Infelizmente, essas grandes taxas de aprendizado conduzem para a oscilação da rede. Assim, deve-se ter um compromisso entre a convergência do algoritmo e o tempo necessário para atingi-la.

Em relação à função de transferência dos neurônios pertencentes a camada de saída, verificou-se que para a função sigmóide obteve-se um resultado inferior do que para a função linear. Isto porque, com esta, a taxa de aprendizado foi diminuída proporcionando, assim, um tempo de convergência menor.

5.3. RESULTADOS DOS TESTES REALIZADOS

Com os resultados descritos até aqui não pôde-se chegar a uma conclusão a respeito de valores exatos para os quais o sistema tenha um comportamento satisfatório, mesmo conhecendo-se a influência dos parâmetros anteriormente citados. Assim, foram

escolhidas algumas configurações para avaliação. Estes valores variam como descrito na Tabela 5.1.

Até o momento, a rede que apresentou eficiência suficiente para ser adotada na operação de controle de uma célula flexível de manufatura foi uma rede com dezesseis neurônios de entrada, sessenta neurônios na camada intermediária e quatro na camada de saída, de acordo com a avaliação da Tabela 5.1. A taxa de erro utilizada para todos os testes foi de 0.00055 e, na média, este erro apresentou o comportamento expresso na Figura 5.6, mostrando convergir para o erro especificado e sem oscilação.

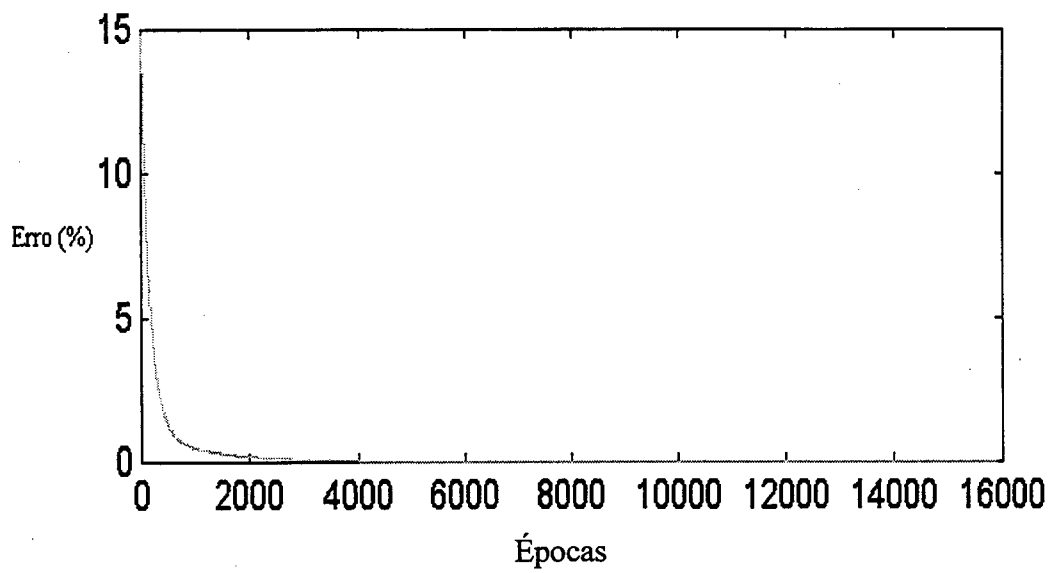

Figura 5.6. Taxa de erro *versus* número de épocas

	Número de neurônios na camada intermedia -ria	Número de padrões para cada palavra	Taxa de erro (%)
16 neurônios de entrada 4 neurônios de saída 2 palavras	0	1	83
		2	87
		3	82
		10	88
	2	1	18
		2	17
		3	11
		10	10
	4	1	17
		2	15
		3	14
		10	14
	6	1	15
		2	14
		3	16
		10	14
	8	1	16
		2	16
		3	15
		10	14
	16	1	17
		2	25
		3	15
		10	16
	25	1	16
		2	14
		3	16
		10	12
	32	1	15
		2	13
		3	11
		10	9
	48	1	7
		2	8
		3	8
		10	5
	60	1	6
		2	3
		3	2
		10	0

Tabela 5.1 Resultados

Com esta rede é possível reconhecer quatro comandos e para cada um deles são necessários dez padrões de treinamento. Desta forma, este sistema apresenta uma taxa de

erro variando de 0 a 15% e trabalha com os seguintes comandos: "abre", "fecha", "liga" e "desliga" (Tabela 5.2).

Palavras	Número de Pronúncias	Taxa de erro (%)
"abre"	40	0
"fecha"	40	5
"liga"	40	0
"desliga"	40	2

Tabela 5.2. Taxa de erro para cada palavra

O tempo de execução de um comando pela célula de manufatura é suficientemente pequeno para ser considerado desprezível, com um atraso imperceptível ao operador humano. O tempo de retorno de cada execução de um determinado comando gira em torno de 1 min, independentemente da tarefa executada.

5.4. SUMÁRIO

Neste capítulo foram apresentados os resultados obtidos no presente trabalho. A princípio, foram analisados os resultados relacionados à variação de alguns dos parâmetros da rede, como número de camadas, número de neurônios nas camadas intermediárias, número de neurônios na camada de entrada, entre outros. A seguir, baseando-se nas análises anteriores, foi definida a configuração final da rede através da realização de testes, cujos resultados são ilustrados por tabelas. Por fim, são apresentados os resultados com relação ao funcionamento geral do sistema implementado.

O próximo capítulo apresentará as conclusões obtidas ao longo deste trabalho, bem como as perspectivas para trabalhos futuros.

Capítulo 6

CONCLUSÕES E PERSPECTIVAS

6.1. INTRODUÇÃO

Neste capítulo serão comentadas as conclusões obtidas deste trabalho. Em primeiro lugar, serão abordadas as questões a respeito do modelo de rede neural adotado, em seguida, aspectos referentes aos procedimentos de aquisição e análise do sinal e, por fim, algumas conclusões sobre o sistema implementado como um todo e sua interação com a célula de manufatura.

6.2. CONCLUSÕES

Este trabalho teve como objetivo o desenvolvimento de um sistema de reconhecimento de comandos verbais, utilizando, para isso, as redes neurais artificiais. A rede implementada foi uma *Perceptron* multi-camadas usando como algoritmo de treinamento o *Backpropagation*. Este sistema foi associado a um robô, que deveria realizar as tarefas correspondentes aos comandos verbais pronunciados por um locutor.

O sistema de reconhecimento de fala foi implementado de modo a ser adaptável ao locutor e mostrou-se bastante eficiente. A estrutura adotada (Figura 2.3), essencialmente modular, possibilita uma grande abstração do sistema, além de uma visão crítica dos módulos e facilidade na resolução dos problemas a eles relacionados.

Em relação à rede neural implementada, uma das primeiras conclusões obtidas é que a área de redes neurais necessita de muitas pesquisas no sentido de definir uma metodologia de projeto de redes. O que se dispõe, atualmente, é de um conjunto de recomendações baseadas em observações, ou seja, totalmente empírico. A continuidade e os avanços das pesquisas nesta área tendem a auxiliar os projetos, no sentido de diminuir o esforço e o tempo despendidos. Por outro lado, as recomendações existentes já poderiam servir de base para um processo automatizado de projeto de redes neurais artificiais.

Outras conclusões, ou confirmações da literatura especializada, foram obtidas neste trabalho. Um exemplo das dificuldades encontradas na etapa de projeto é a escolha dos dados do conjunto de treinamento. Para isso, é necessário um estudo criterioso para escolher os padrões que irão representar o conjunto de treinamento, sendo que, pelos resultados do Capítulo 5, pôde-se observar que quanto maior o número de padrões, menor é a taxa de erro. Por outro lado, quanto maior o número de padrões, maior o tempo de treinamento.

Em relação ao número de neurônios na camada intermediária, foi comprovado que elevando-se o número destes promove-se o decréscimo da taxa de erro. Por outro lado, o número elevado de neurônios pode inviabilizar a utilização da rede, por questões de memória e de tempo de processamento. Portanto, estas questões têm que ser consideradas no projeto. Por sua vez, a taxa de aprendizado deve ser pequena de forma que a rede possa convergir sem apresentar oscilação.

O algoritmo de treinamento utilizado, *Backpropagation,* apesar de não ter sido sujeito a nenhuma espécie de tratamento com o intuito de diminuir-se o tempo de convergência e nem o problema do mínimo local, foi adequado e proporcionou um resultado satisfatório.

Outro fator comprovado foi a dificuldade em trabalhar-se com dados de fala. Isto dificultou o processo de escolha do conjunto de treinamento bem como do modelo de rede para executar o reconhecimento.

O processo de aquisição e análise do sinal é uma etapa bastante complexa. O tempo despendido nesta etapa é igual ou até mesmo superior aquele da etapa de classificação. Dentre os problemas encontrados e as soluções adotadas, pode-se destacar que a não utilização de um *buffer* intermediário para o armazenamento do sinal de entrada mostrou-se bastante satisfatória. O *buffer* intermediário é utilizado para possibilitar uma alta taxa de amostragem do sinal mas, com o modo de transferência direto, descrito no Capítulo 4, o sistema conseguiu manter uma alta taxa de amostragem.

Em relação à análise do sinal, a medida de adotar-se a detecção de início e fim de uma palavra foi de suma importância. Sem este procedimento o reconhecedor teria um desempenho degradado, porque trabalharia sobre um tamanho fixo de janelas e, provavelmente, seria difícil que uma palavra tivesse seu início e fim sincronizados com o início e fim da janela. Com a detecção de início e fim de palavras, a probabilidade de reconhecimento é maior, resultando, consequentemente, em economia de memória e de tempo de processamento.

Quanto à análise do sinal segmentado, conclui-se que a mesma possibilita uma economia de memória significativa, resultando diretamente na possibilidade de implementação de redes com maiores dimensões sem que haja necessidade de estratégias adicionais, tais como manipulações de dados em arquivo, expansão de *hardware*, entre outras.

Em relação a linguagem de programação utilizada, pode-se concluir que C pôde, de forma eficiente, suprir as necessidades de implementação presentes neste trabalho. A técnica de programação seqüencial foi adequada, resultando num tempo de reconhecimento, pelo sistema, bastante pequeno, em torno de milisegundos.

A taxa de erro alcançada pelo o sistema de reconhecimento foi bastante satisfatória, girando em torno de 10 %. Estes resultados são comparáveis aos do sistema *Voicetype*, desenvolvido pelo IBM em 1994. Neste sistema são reconhecidas quatro palavras com uma taxa de erro variando de 10 a 20% [Taf].

O sistema não sofre nenhuma degradação em seu funcionamento, mesmo quando aliado à célula de manufatura, mostrando-se robusto aos ruídos ambientes nos quais foi desenvolvido. Assim sendo, este sistema como um todo apresenta a mesma taxa de erro que é encontrada somente devido ao sistema de reconhecimento de fala.

De modo geral, pôde-se concluir a viabilidade desta implementação. Acredita-se que este sistema possa ser largamente utilizado em aplicações pertencentes aos mais diversos domínios.

6.3. PERSPECTIVAS DE TRABALHOS FUTUROS

Em relação às perspectivas de futuros trabalhos nesta área, ou como continuação do trabalho apresentado, podem ser citadas:

- O aumento do número de comandos identificados pelo sistema;
- Variação da técnica de análise do sinal de fala, variação da arquitetura de rede, entre outros, a fim de avaliar e comparar o desempenho deste sistema;
- Testar outros algoritmos de treinamento;
- Implementação da rede neural artificial utilizando-se técnicas de programação paralela;
- Introduzir novas gramáticas.

BIBLIOGRAFIA

[All] Allen Bradley Co. ***Manuais***, 1993.

[Ant] Antunes, A. C. S. and Carrijo, G. A. ***"Uso de Quantização Vetorial e Mapas de Kohonen para Traçado de Trajetórias do Sinal de Fala"***. Anais do I CBRN, Itajubá. Outubro. 1994.

[Bar] Barreto, J. M. ***"Redes Neurais: Fundamentos e Aplicações"***. Curso oferecido no II SBAI, Curitiba. Setembro. 1995.

[Bez] Bezdek, J. C. ***"Pattern Recognition with Fuzzy Objective Function Algorithms"***. Plenum Press. 1987.

[Bor] Borland C++ 4.5 ***Manuais***.

[Caj] Cajal, S. ***"A New Concept of the Histology of the Central Nervous System"***. In Rottenberg and Hochberg. Neurological Classics in Modern Translation. New York, Hafner. 1977.

[Cho] Chomsky, N. ***"Three Models for the Description of Language"***. IRE Trans. Inform. Theory. N. 3. Pp. 113-124. 1956.

[Coo] Cooley, J. W. and Tukey, J. W. ***"An algorithm for the machine calculation of complex Fourier series"***. Mathematics of Computation. N 19. Pp. 297-301. 1965.

[DKS] ***"Developer Kit for Sound Blaster Series"***. Hardware Programming Reference. Second Edition.

[Ell] Elliott, D. F. ***"Fast Transforms Algorithms, Analyses, Applications"*** . Academic Press, Inc. 1982.

[Fan] Fant, G. ***"Speech Sounds and Features"***. Massachusetts Institute of Technology. MIT Press. 1973.

[Fre] Freeman, J. A. and Skapura, D. M. ***"Neural Networks: Algorithms, Applications and Programming Techniques"***. Addison Wesley, 1991.

[Fu] Fu, K. S. ***"Application of Pattern Recognition"***. CRC Press, Cleveland, OH. 1982.

[Ged] Geddes, K. O. ***"Algorithms for computer algebra"***. Kluwer Academic Publishers. 1992.

[Hay-a] Haykin, S. ***"Advances in Spectrum Analysis and Array Processing"***. Prentice Hall Advanced Reference Series. Prentice Hall. 1991.

[Hay-b] Haykin, S. ***"Neural Networks"***. Canadá, Toronto. McMaster University. 1994.

[Hop] Hopfield, J. J. ***"Neurons with graded response have collective computational properties like those of two-state neurons"***. Proceedings of the National Academy of Sciences U.S.A., 81. Pp. 3088-3092. 1984.

[Kar] Karayiannis, N. B. and Venetsanopoulos, A. N. ***"Artificial Neural Networks: Learning Algorithms, Performance Evaluation, and Application"***. Kluwer Academic Publishers. 1993.

[Koh-a] Kohonen, T. ***"Self-Organization and Associative Memory"***. Vol. 8 of Springer Series in Information Sciences. Springer-Verlag. New York. 1984.

[Koh-b] Kohonen, T. ***"The "neural" phonetic typewriter"***. Computer, 21(3). Pp. 11-22. March. 1988.

[Law] Lawrence, S., Tsoi, A. C. and Back, A. D. ***"The Gamma MLP for Speech Phoneme Recognition"***. Advances in Neural Information Processing Systems. MIT Press. 1996.

[Lea] Leal, A. B. ***"Relatório de Atividades"***. Labortaório LCMI-Lai. Março. 1997.

[Lip-a] Lippman, R. ***"An Introduction to Computing with Neural Nets"***. IEEE Computer Society. Vol. 3, n. 4. Pp. 4-22. April, 1987.

[Lip-b] Lippman, R. and Gold, B. ***"Neural Classifiers Useful for Speech Recognition"***. In 1st International Conference on Neural Networks, IEEE. 1987.

[Lip-c] Lippman, R. ***"Review of Neural Networks for Speech Recognition"***. Neural Computation 1(1). Pp. 1-38. 1989.

[Lip-d] Lippman, R.. ***"Pattern classification using neural networks"***. IEEE Communications Magazine. Pp. 47-64. 1989.

[Mar] Marcel, H. and Luna, P. T. L. ***"Utilização de Redes de Kohonen no Projeto RAFA"*** Dynabis. Blumenau. FURB. Vol. 1, n. 5. Dezembro 1993.

[McC] McCulloch, W. S. and Pitts, W. H. ***"A Logical Calculus of the Ideas Immanent in Nervous Activity"***. Bull. Math. Biophys., 5. Pp. 115-133. 1943.

[Nel] Nelson, P. A. and Elliott, S. J. ***"Active Control of Sound"***. Academic Press. 1992.

[Pra] Prager, R., Harrison, T. and Fallside, F. ***"Boltzmann Machines for Speech Recognition"***. Computer Speech and Language 1. Pp. 2-27. 1986.

[Pet] Petzold, C. ***"Programming Windows 95".*** Microsoft Press. 1996.

[Red] Reddy, D. R. ***"Speech Recognition"*** . Invited Papers of the IEEE Symposium. USA: Mit Press. 1973.

[Ros] Rosenblatt, F. ***"The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain".*** Psychological Rev., 65. N. 6. Pp. 386-408. 1958.

[Rum-a] Rumelhart, D. E., McClelland, J. L. and PDP Group. ***"Parallel Distributed Processing".*** MIT Press, Vol. I & II. Cambridge, Massachusetts. 1986.

[Rum-b] Rumelhart, D. E. et al, ***"Integrating Neural Networks into Computer Speech Recognition Systems".*** GOMAC 92.

[Sch] Schalkoff, R. ***"Pattern Recognition. Statistical, Structural and Neural Approaches".*** John Wiley & Sons, Inc. 1992.

[Sep] Sepeda, I. H. Fo. ***"Desenvolvimento e Implementação de um sistema de reconhecimento de peças baseado em Redes Neurais".*** Dissertação de Mestrado. Curso de Pós-Graduação em Eng. Elétrica - UFSC. 1994.

[Skl-a] Sklansky, J. ***"Pattern Recognition: Introduction and Foundations".*** Dowden, Hutchinson & Ross, Inc. 1973.

[Skl-b] Sklansky, J. ***"Threshold Training of Two-Mode Signal Detection".*** IEEE Trans. Inform. Theory. N. 3. Pp. 353-362. 1965.

[Sou] Soucek, B. ***"Neural and Concurrent Real-Time Systems".*** Sixth-Generation Computer Technology Series. Wiley-Interscience. 1989.

[Taf] Taffner, M. A. ***"Reconhecimento de Palavras Faladas Isoladas Usando Redes Neurais Artificiais".*** Dissertação de Mestrado. Eng. de Produção. 1996.

[Uhr] Uhr, L. ***"Pattern Recognition, Learning and Thought"***. Prentice Hall. 1973.

[Wai-a] Waibel, A. and Lee, K. ***"Reading in Speech Recognition"***. Morgan Kruffmman. USA, Califórnia. 1990.

[Wai-b] Waibel, A. et al, ***"JANUS: A Speech-to-Speech Translation System Using Connectionist and Symbolic Processing"***. ICASSP, 1991.

[Wai-c] Waibel, A. et al, ***"Flexibility Through Incremental Learning: Neural Networks for Text Categorization"***. Proceeding of the World Congress on Neural Networks (WCNN) 1993, Portland, Oregon, pp. 24-27, July, 1993.

[Wid-a] Widrow, B. and Winter, R. ***"Neural Nets for Adaptative Filtering and Adaptative Pattern Recognition"***. IEEE Computer. March. 1988.

[Wid-b] Widrow, B. and Lehr, M. A. ***"30 years of adaptative neural networks: Perceptron, Madaline, and Backpropagation"***. Proceedings of the IEEE, 78. Pp. 1415-1442. 1990.

[Wis] Wisbeck, J. O. ***"Técnicas Híbridas de Processamento de Sinais Biomédicos Implementadas com Redes Neurais Artificiais"***. Exame de Qualificação ao Doutoramento. GPEB - UFSC. 1997.

[You] Young, S. ***"A Review of Large-vocabulary Continuous-speech Recognition"***, IEEE Signal Processing Magazine. Pp. 45-57. September, 1996.

[Zad] Zadeh, L. et al. ***"Fuzzy Sets and Their Applications to Cognitive and Decision Processes"***. Academic Press. New York. 1975.

[Zil] Ziller, R. ***"CONCEL versão 1.1"***, Laboratório LCMI-LAI, Agosto, 1994.